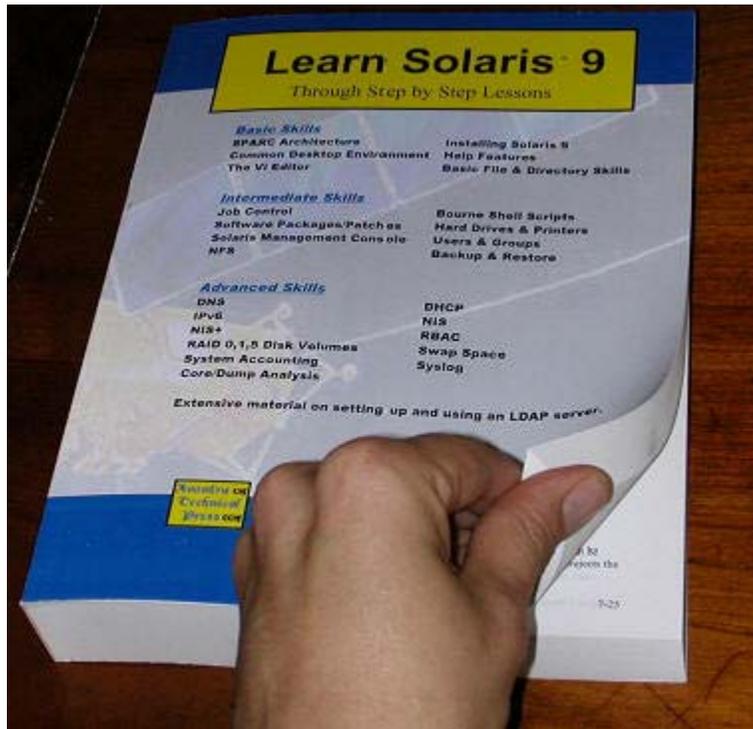


Please consider purchasing the paperback book or giving a donation.



If you lost your job then please do not feel obligated to pay for a paperback book or send in a donation. In all honesty I would much rather have you spend thirty dollars on your family or your bills than buy my book. If you can give a donation or pay for a paperback book I would really appreciate your contribution. Understand that this book took me about one year of time and \$3,500 to produce. I do need some source of funding to produce the e-books. If you live in a foreign country where shipping the book would be prohibitive, maybe you could consider giving a donation to support this e-book.

Steven A. Beebe
Author
Avontra Technical Press

<http://www.teachmesun.com> for purchase or donation details.

What did it take to produce this computer book?

The original text took me about six months of full time work to complete. After the original text was completed, I spent an additional month checking the lessons and performing the initial grammar cleanup. I then hired a very good technical/copy editor to work on the book's format and layout for another two months. Once the technical editing and copy editing was completed it took me an additional two weeks to setup the margins, page numbering, picture/table labels, table of contents and the index.

After the book's internal contents were created, I setup a print run of four test proofs. The bindery (process of putting a cover on the book) took an additional two weeks before the final book layout looked really nice.

If you ever pickup a computer book, look at all the staff involved with producing a modern computer book. I had to do almost all of these functions by myself. In the end, I did hire some other people to take care of some things that I did not have experience with.

As for the pure raw economic cost, I would estimate I have spent \$3,500 so far. Let's not even go into the wages I lost if I was not trying to produce a computer book for a living :(

Some people have asked me, if you put that much time, money and effort into the book, why are you giving away the e-book for free? Are you completely nuts?

Initially I did charge people for the e-book out of Content Reserve. Then one night I had this crazy idea of just giving away the e-book to anyone that needed it and could not pay. If someone lost their job or is low on cash, I will let them use the book's contents for free. I am eventually hoping that other people that can pay for a paperback book will purchase the book to keep the funding alive.

I am human, I understand that there are also a lot of people that have lost their jobs and are struggling to pay the bills. Most college students (especially students in India and Africa) have a hard time paying for American computer books. Maybe my idea of giving away the book and depending on donations is crazy! who knows?

If you would like to receive a paperback book or send a donation please send me an email message :

author@teachmesun.com

One obvious question:

How do I know I am not sending my money to some con-man on the internet that just copied the file from someone else? Understand that a PDF file is basically an electronic picture of a document. This PDF file should have a background diagonal ghost image

Only Buy From --- <http://www.teachmesun.com>

This should appear on the first page of chapters 1 through 7 (and some other pages) chosen randomly. I control the domain name [teachmesun.com](http://www.teachmesun.com). I believe it would be rather difficult to remove the slanted ghost text from a PDF file. Imagine trying to remove the background image of clouds from a Picasso without anyone noticing? I am under the belief that this can not be done? This whole Solaris book thing has been complete and absolute financial disaster for me.. What con-man would be dumb enough to risk a felony conviction for the trickle of money that this financially God forsaken idea has given me already?

Copyright Restrictions Explained

1. You are free to copy and redistribute this e-book with the following copyright provisions :
 - a. You may distribute this e-book for free to the general public.
 - b. You may not alter the file's electronic contents or format.
 - c. You may not change the file's name "Free-Solaris-Ebook.zip" If you want to post or distribute the file, you must distribute the file under the name "Free-Solaris-Ebook.zip" to be legal. Do not post the file in PDF format or another format. Keep in mind on thing, if you post the file as a PDF on the internet, people will bookmark the file like a webpage. Each time they visit the file, your internet server will have to download a seven MB file each time. Keep the file as a ZIP file so visitors will be more inclined to download the file instead of book marking the file.
2. You may print page(s) from the file's contents onto 8.5" x 11" paper for your personal use.
3. If you want to reference the text, concepts or images from this e-book please contact the author (Steven A. Beebe, Avontra Technical Press) for permission. author@teachmesun.com. Express written permission from Steven A. Beebe is required to use's this book's contents.

Publisher

Avontra Technical Press <http://www.teachmesun.com>
12690 Falcon Drive
Colorado Springs, CO 80908
sales@teachmesun.com

For information on book distributions outside the U.S.A. or to arrange bulk purchase discounts, sales contracts or other promotional/distribution ideas, please contact Avontra Technical Press via email sales@teachmesun.com or postal letter. Please visit this book's official website <http://www.teachmesun.com> for additional information.

Copyright Notice – Copyright @ 2003 Avontra Technical press, All rights Reserved.

Avontra Technical Press eBooks are copyrighted works and may not be converted to any other format, electronic or otherwise, nor sold or distributed without the prior written permission from Avontra Technical Press. This eBook “Learn Solaris 9 Through Step by Step Lessons” (including but not limited to the text, images, audio, and/or video) may be electronically copied and reproduced, uploaded, posted, transmitted, or electronically distributed in any way so long as the file's original name “Free-Solaris-Ebook.zip” and contents remain in tact. The eBook copy may be downloaded for your personal, non-commercial use on to a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice. This eBook contains trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties. Obtaining this eBook, whether it be in print or electronically, does not give the end user any of the rights granted to the copyright holder. The copyright owner has the exclusive right to: To prepare derivative works based upon the work; To distribute copies of the work to the public by sale or other transfer of ownership, or by rental, lease, or lending. Avontra Technical Press conducts thorough digital investigations (through internet sniffing programs) for any suspicious web activity, sales, or auctions that may infringe on the copyrights of Avontra Technical Press. Once a copyright violation is established, Avontra Technical Press will take legal action against any offending individual or business entity. This action includes, but is not limited to hosting companies, payment gateways, and other companies that may be involved, No patent liability is assumed with respect to the use of the information contained herein. The publisher and author assume no responsibility for errors and omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Avontra Technical Press can not guarantee that every trademark or service mark is perfect. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark. Solaris is a trademark of Sun Solaris. Windows is a registered trademark of Microsoft Corporation. MS-DOS is a registered trademark of Microsoft Corporation.

Warranty of Fitness and Disclaimer

Every effort has been taken to make this book complete and accurate, but no warranty of fitness is implied. The information is provided is on an “as is” basis. The author and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of any program or CD-ROM(s) shipped with this book or downloaded from the book's official webpage(s).

Intended Audience

This book is intended for junior to mid-level system administrators. Ideally, readers should have access to Solaris 9 and a SPARC-based system. Most lessons can also be performed with Solaris 7 and 8. This book can also be used with HCL compatible Intel-based systems.

Readers who have the Intel or SPARC version of Solaris 7 or Solaris 8 can still perform many of the lessons in this book. However, there are some lessons that specifically deal with new features of Solaris 9 that the reader can not perform with previous versions of Solaris. Please read the next section, "Recommended Solaris Versions and Hardware."

The reader needs to have root access on the test workstation. This workstation should not be used for mission-critical applications or for live production traffic. Try to use a SPARC 5 workstation or a SunBlade 100 workstation. Other higher performance SPARC systems can also be used for the lessons.

Recommended Solaris Versions and Hardware

To perform all the lessons in this book, the reader needs to have access to a SPARC-based system and Solaris 9. Sun Microsystems is planning on releasing an Intel version of Solaris 9 sometime in the year 2003. Please note that the lessons that deal directly with SPARC hardware cannot be performed on an Intel-based system. Most of the lessons in this book can be performed on the SPARC- and Intel-based versions of Solaris 7 and Solaris 8. Some lessons that deal specifically with Solaris 9 features obviously can not be performed with Solaris 7 or Solaris 8.

Solaris 9 can be purchased from <http://sunstore.sun.com>. New SPARC-based systems can be purchased directly from Sun Microsystems, or used SPARC-based systems can be purchased from online auction sites like <http://www.ebay.com> or <http://auctions.shopping.yahoo.com>.

Please Visit Our Website

<http://www.teachmesun.com>

This website has additional material used in some of the lessons. This website also has additional information about other books and training material in progress. It also has file uploads, promotions, and a calendar of trade shows and professional events that we are involved with.

- Interested in a book proposal?
- Want to become a reseller of this book?
- Have comments and information about this book? Please report any errors that you may find.
- Are you an expert in Solaris or Cisco®? Would you like to freelance as a reviewer or technical writer?
- Do you have a proposal that might produce a good partnership on a project?

Table of Contents

CHAPTER 1 WORKING WITH SPARC SYSTEMS	1-1
How to Purchase an Ultra 5 Workstation.....	1-2
How to Purchase a SunBlade 100 Workstation	1-4
Solaris 9 Media Kits	1-4
Solaris 9 Slim Kit	1-6
Solaris 9 System Administrator's Kit.....	1-7
Early Access Solaris 9	1-8
Licensing Solaris 9	1-8
How to Use a SPARC System	1-10
OpenBoot Acts Different Than a BIOS.....	1-10
Sun Keyboards	1-11
The OpenBoot Prompt.....	1-12
Using a Null Modem Cable.....	1-13
The PROM Chip.....	1-16
The NVRAM Chip.....	1-16
Working with Sun SPARC Equipment	1-17
Some Other Useful PROM Command.....	1-19
NVRAM Variables.....	1-21
Creating Devices Using the nvalias Command.....	1-22
Using the nvedit Command	1-23
The eeprom Command	1-24
Using the Obdiag Utility	1-24
CHAPTER 2 INSTALLING SOLARIS 9	2-1
Solaris 9 Installation Methods.....	2-2
Solaris 9 Media Kits	2-3
Pre-installation Information	2-3
Displaying the OK Prompt.....	2-5
Shutting Down Solaris 9 Gracefully.....	2-38
An Additional Note.....	2-40
Key Points to Remember	2-40
CHAPTER 3 THE COMMON DESKTOP ENVIRONMENT.....	3-1
The Login Manager.....	3-2
The CDE Desktop.....	3-3
Using the Front Panel.....	3-3
Workspace Buttons	3-6
Renaming Workspace Buttons	3-7
Locking the Workstation	3-8
Starting a Command Line Session	3-9
The Workspace Menu	3-9
Shut Down Solaris 9 Gracefully	3-10
The init command	3-10
The shutdown Command.....	3-11
The halt, reboot and poweroff Commands	3-12
How to Suspend the System.....	3-13
CDE Applications	3-13
Help Manager	3-14
Solaris Documentation.....	3-16
Obtaining Adobe Acrobat Reader For Solaris	3-17
File Manager	3-18
Basic Navigation within the File Manager.....	3-18
Creating a New Folder	3-19

Deleting a Folder	3-19
Using File Manager on CDROMs and DVDs	3-19
Using File Manager with Floppies.....	3-19
How to Format a Floppy Disk with File Manager	3-20
Command Line Tools for Floppy Disks	3-21
Using the Find File Utility.....	3-22
Using the File Properties Utility.....	3-24
Using the Calculator Utility.....	3-26
The Netscape Navigator Browser Version 4.87	3-27
Using the Text Editor	3-27
The Style Manager.....	3-29
Understanding the System_Admin Folder Utilities	3-31
Creating a Custom Made Icon	3-33
Administration of the CDE.....	3-35
Administration of the Login Manager.....	3-35
Administration of the Session Manager	3-35
Understanding the CDE File structure.....	3-36
CHAPTER 4 WORKING WITH FILES AND DIRECTORIES	4-1
Working with Directories	4-2
Using the vi text editor	4-4
Examining Text Files.....	4-16
CHAPTER 5 GETTING HELP	5-1
Online Documentation	5-2
The http://docs.sun.com Webpage	5-2
The http://sunsolve.sun.com Webpage	5-3
The http://www.sun.com/bigadmin Webpage	5-3
Using Man Pages.....	5-4
Adding Man Pages to a System.....	5-7
CHAPTER 6 SOLARIS BOOTUP AND SHUTDOWN	6-1
The Four Phases of the Boot Process	6-1
Phase 1 The Boot PROM Phase	6-1
Phase 2 Boot Program Phase.....	6-3
Phase 3 Kernel Phase.....	6-3
Phase 4 The Init Phase	6-4
Solaris Run Levels.....	6-4
Run Control Scripts	6-5
Understanding Run Levels.....	6-5
Examples of the /etc/inittab File and Run Levels.....	6-6
Example Scripts from the /etc/rc0.d directory.....	6-7
CHAPTER 7 WORKING WITH USERS AND GROUPS	7-1
Files Related to Users	7-2
The /etc/passwd file	7-2
The /etc/group file.....	7-3
The /etc/shadow file	7-4
A Closer Look at Solaris Users.....	7-4
The Login Shell	7-5
The Initialization File	7-5
Creating User Accounts.....	7-5
A Closer Look at the /etc/passwd, /etc/group and /etc/shadow Files.....	7-6
The /etc/passwd file	7-6
The /etc/group File.....	7-7
The /etc/shadow File	7-8
Permissions on Files and Directories with Users.....	7-9

Directory Permissions	7-9
File Permissions	7-9
The <code>chmod</code> Command	7-10
Groups Within Solaris 9	7-15
Using the Solaris Management Console (SMC)	7-18
Creating Users from Templates	7-21
How Name Services Work with Users and Groups	7-29
Quota Limits.....	7-30
Terms Used with Quotas	7-30
Setting Quota Limits.....	7-30
Commands Used with Quotas	7-31
Checking Quotas with <code>repquota</code>	7-34
Changing a User's Quota Limits.....	7-35
Disabling a File System's Quota Limits	7-36
CHAPTER 8 WORKING WITH DIRECTORIES AND FILES	8-1
The <code>/dev</code> and <code>/devices</code> Directories	8-1
The <code>/bin</code> Directory	8-2
The <code>/etc</code> Directory	8-3
The <code>/export</code> Directory	8-4
The <code>/home</code> Directory.....	8-4
The <code>/kernel</code> Directory.....	8-4
The <code>/mnt</code> Directory	8-4
The <code>/opt</code> Directory	8-5
The <code>/sbin</code> Directory.....	8-5
The <code>/tmp</code> Directory	8-5
The <code>/usr</code> Directory	8-5
Types of Files.....	8-6
Understanding Device Files	8-6
Understanding Hard Links	8-7
Understanding Soft Links.....	8-7
What Is a Directory?	8-9
What Are Regular Files?.....	8-9
Using the <code>ls</code> Command	8-9
CHAPTER 9 WORKING WITH SOFTWARE PACKAGES.....	9-1
Software Packages.....	9-2
Software Management Command Line Utilities	9-3
The <code>pkgadd</code> Command	9-3
The <code>pkgrm</code> command.....	9-4
The <code>pkgchk</code> command	9-4
The <code>pkgparam</code> command.....	9-5
Spooling a Software Package	9-6
Using GUI Software Package Management Tools	9-7
Admintool.....	9-7
The Solaris Product Registry.....	9-9
Solaris Web Start Program.....	9-11
CHAPTER 10 WORKING WITH PATCHES.....	10-1
Why Have Software Patches?	10-1
The <code>patchadd</code> command.....	10-2
The <code>/var/sadm/patch</code> Directory.....	10-3
The <code>patchrm</code> Command.....	10-5
Patch Reports	10-5
Patch Structure.....	10-6
Using Patch Manager	10-6

CHAPTER 11 JOB CONTROL.....	11-1
Using the <code>crontab</code> command.....	11-1
Format Rules and Examples of the Crontab Command.....	11-2
The <code>cron.allow</code> and <code>cron.deny</code> files.....	11-3
The <code>at</code> command.....	11-4
Process Control.....	11-6
The <code>prstat</code> Command.....	11-7
CHAPTER 12 WORKING WITH SHELL SCRIPTS	12-1
Bourne Shell.....	12-3
Command.....	12-6
Permissions Set By the Command.....	12-6
Display of Permissions.....	12-6
Using Variables.....	12-6
Reading Values into a Script.....	12-10
Command Line Arguments.....	12-11
Mathematical Expressions in the Bourne Shell Script.....	12-14
Using Functions in a Bourne Script.....	12-15
Control Statements.....	12-18
The <code>if</code> control statement.....	12-18
Using the <code>else</code> option with the <code>if</code> Command.....	12-20
Creating an <code>if/else if</code> Tree.....	12-20
Other Useful <code>if</code> Test Conditions.....	12-21
The <code>case</code> Command.....	12-22
Creating a Selection Menu with the <code>Case</code> Command.....	12-25
The <code>for</code> Command.....	12-25
Using the <code>while</code> Command for Numerical Loops.....	12-29
The <code>Until</code> Command.....	12-30
The <code>break</code> command.....	12-31
The <code>continue</code> command.....	12-31
Debugging Bourne Shell Scripts.....	12-32
Bourne Run Control Scripts.....	12-34
Users <code>.profile</code> File.....	12-35
CHAPTER 13 WORKING WITH HARD DRIVES.....	13-1
Hard Drive Slices.....	13-1
The Swap Partition Slice 1 and Disk Image Slice 2.....	13-2
What Makes Up a Hard Drive and How That Relates to Solaris 9.....	13-3
Figure 13.6 Multiple Platters and a Spindle.....	13-6
Figure 13.7 Track and Cylinder.....	13-6
Solaris Disk Naming Conventions.....	13-9
Logical and Physical Device Names.....	13-10
The <code>mount</code> Command.....	13-11
OPTIONAL.....	13-13
CHAPTER 14 USING THE MOUNT AND SHARE COMMANDS ...	14-1
Understanding Mounting.....	14-1
How to Unmount a Busy or Jammed File System.....	14-4
Accessing Removable Media.....	14-5
Special Mount Options.....	14-6
The <code>/etc/vfstab</code> File.....	14-7
The <code>/usr/sbin/umountall</code> Command.....	14-9
Mounting Different Media.....	14-9
Sharing a Directory on a Network.....	14-10
Common Terms.....	14-10
Using the <code>share</code> Command.....	14-10

Using Share Access Lists	14-11
Using the <code>unshare</code> Command	14-13
The <code>shareall</code> and <code>unshareall</code> commands.....	14-13
Figure 14.3 The <code>/etc/dfs/dfstab</code> File.....	14-14
The <code>dfshares</code> Command	14-14
Special Network Mounting Options.....	14-14
CHAPTER 15 SOLARIS FILE SYSTEMS	15-1
Local File Systems.....	15-1
Distributed File Systems	15-1
Networked File Systems.....	15-1
Pseudo File Systems	15-2
UFS Disk Structure and the File System Structure.....	15-2
File System Components in Detail	15-5
The SuperBlock	15-5
Cylinder Group Block.....	15-6
The Inode Table	15-6
Data Blocks and Fragment Blocks	15-7
Monitoring a File System	15-8
The <code>du</code> Command.....	15-9
The <code>quot</code> Command.....	15-10
The <code>fsck</code> Command.....	15-11
The <code>newfs</code> Command	15-12
CHAPTER 16 TCP/IP ON SOLARIS	16-1
Common Terms	16-1
The OSI Model and TCP/IP Model	16-3
The OSI model	16-3
Application Layer Programs	16-7
Other Network Concepts.....	16-7
The TCP/IP Model.....	16-9
OSI MODEL TCP/IP Model.....	16-9
Networking Files on Solaris 9	16-9
Understanding the <code>ifconfig</code> Command	16-10
The <code>banner</code> Command.....	16-12
The <code>ping</code> Command.....	16-13
TCP Ports.....	16-14
The <code>inetd</code> daemon	16-14
RPC (Remote Procedure Calls)	16-15
The <code>snoop</code> Command	16-16
CHAPTER 17 PRINTING IN SOLARIS	17-1
Printing Concepts	17-2
Some Common Terms Used with Printing in Solaris 9.....	17-2
Important Directories, Files, and Commands for Solaris 9 Printing.....	17-3
Other important files and directories are:.....	17-3
Understanding the <code>/usr/share/lib/terminfo</code> Directory.....	17-4
Solaris GUI Printing Tools.....	17-5
Command Line Print Management Tools.....	17-11
Alternate Command Line Print Commands	17-11
How Solaris Finds the Default Printer	17-12
CHAPTER 18 BACKUP AND RESTORE	18-1
The <code>ufsdump</code> and <code>ufsrestore</code> Commands.....	18-2
Tape Drive Locations.....	18-5
The <code>ufsdump</code> Command.....	18-5

Examples of the <code>ufsdump</code> Command	18-6
Types of Tapes.....	18-6
The <code>ufsrestore</code> command.....	18-6
DUMP: Date of this level 0 dump: Tue 10 Sep 2002 09:27:08 PM MDT.....	18-8
DUMP: Date of last level 0 dump: the epoch.....	18-8
DUMP: Dumping (Pass IV) [regular files].....	18-8
The <code>tar</code> command.....	18-10
Compressing, Viewing and Uncompressing Files	18-12
The <code>zip</code> and <code>unzip</code> Utilities.....	18-13
The <code>compress</code> and <code>uncompress</code> utilities	18-15
The <code>zcat</code> utility.....	18-15
The <code>dd</code> Command.....	18-16
CHAPTER 19 THE SOLARIS MANAGEMENT CONSOLE	19-1
Starting the SMC	19-1
SMC Components.....	19-3
Using the SMC	19-4
The Tool Bar	19-6
The Location Bar	19-7
The Navigation Pane	19-7
The View Pane	19-8
The Help Pane	19-8
Using the SMC	19-9
Solaris System Status	19-9
Solaris Process Manager 1.1.....	19-10
Log Viewer 4.1	19-12
The Solaris Performance Manager 1.0.....	19-13
Solaris Project Database Manager 1.0.....	19-16
CHAPTER 20 WHAT'S NEW IN SOLARIS 9	20-1
Security Improvements	20-1
Solaris Secure Shell (SSH).....	20-1
Internet Key Exchange (IKE)	20-2
Improved LDAP (Lightweight Directory Access Protocol) Security.....	20-2
128-bit Encryption.....	20-2
Kerberos Key Distribution Center (KDC)	20-2
Improved Process Controls.....	20-2
Fixed-Priority Scheduling.....	20-3
Solaris Resource Manager.....	20-3
Network Compatibility:	20-3
Linux Support.....	20-3
Sendmail 8.12.....	20-3
Mobile Internet Protocol.....	20-4
BIND 8.2.4	20-4
Solaris PPP 4.0.....	20-4
Windows 2000 Support	20-4
What Happened to the GNOME 2.0 Desktop?	20-4
Disk Management	20-5
Ability to Burn CDRoms.....	20-5
Solaris Volume Manager	20-5
New Install Methods.....	20-5
Solaris Live Upgrade.....	20-5
Web Start Flash	20-6
Other Interesting Improvements	20-6
Netscape Navigator	20-6
iPlanet LDAP Directory Server Integration	20-6

SunScreen 3.2.....	20-7
Solaris Freeware.....	20-7
End of NIS+.....	20-7
DVD Installation Disk.....	20-7
Solaris Web Start Wizards SDK 3.0.1.....	20-7
Graphical Workspace Manager Upgrade.....	20-7
Software Development Libraries.....	20-8
WBEM.....	20-8
Improved Solaris Management Console.....	20-8
Human Readable Output.....	20-9
Improved Software Support.....	20-9
Extra Value Software.....	20-9
Freeware.....	20-9
CHAPTER 21 JUMPSTART INSTALL METHODS.....	21-1
The re-prinstall Script.....	21-2
Using the Factory JumpStart Install Method.....	21-4
Using the Custom Jumpstart Installation Method.....	21-4
Common Custom Jumpstart Terms.....	21-4
Custom JumpStart Scripts and Files.....	21-5
Understanding Jumpstart Installations.....	21-5
Performing a Local JumpStart.....	21-7
Local JumpStart Steps.....	21-7
Understanding the rules file.....	21-8
Understanding a Profile Text File.....	21-10
Profile File Keywords.....	21-11
Using the pfinstall Utility to Check a Profile.....	21-17
Understanding the sysidcfg File.....	21-18
Performing a Network JumpStart.....	21-22
The setup_install_server script with the -b Option.....	21-25
Creating a JumpStart Directory.....	21-26
The add_install_client script.....	21-27
The sysidcfg file.....	21-27
Files Used with a Network JumpStart Installation.....	21-28
The /etc/dfs/dfstab File.....	21-28
The /etc/inetd.conf file.....	21-29
The /etc/nsswitch.conf File.....	21-29
The /etc/hosts File.....	21-30
The /etc/ethers file.....	21-30
Some Final Comments on the Networked JumpStart Installation.....	21-32
Networked JumpStart.....	21-32
CHAPTER 22 ROLE BASED ACCESS CONTROL.....	22-1
Key RBAC Concepts.....	22-1
Understanding Rights.....	22-2
Understanding Roles.....	22-4
Understanding Authorization.....	22-4
Understanding Profiles.....	22-4
Understanding Execution Attributes.....	22-5
Commands Used with Roles.....	22-5
Databases Associated with RBAC.....	22-6
/etc/security/exec_attr.....	22-6
/etc/security/auth_attr.....	22-6
/etc/user_attr.....	22-6
/etc/security/prof_attr.....	22-6

The <code>user_attr</code> database file.....	22-6
The <code>auth_attr</code> database file.....	22-7
The <code>prof_attr</code> database file	22-7
The <code>exec_attr</code> database file	22-8
CHAPTER 23 NIS AND NIS+	23-1
Common NIS Terms	23-2
NIS Maps.....	23-3
How NIS Map Files Are Named.....	23-3
Common NIS Commands.....	23-5
The NIS Client	23-9
How to Remove NIS.....	23-11
NIS+	23-13
NIS+ Tables.....	23-14
NIS+ Name Space	23-14
NIS+ Table Updates	23-16
Overview of the NIS+ Commands.....	23-17
The <code>nismaster</code> Command.....	23-17
Working with NIS+ Tables.....	23-20
The <code>niscat</code> command	23-20
CHAPTER 24 WORKING WITH SYSLOGD.....	24-1
Syslog Error Messages.....	24-1
The <code>/etc/syslog.conf</code> File in Detail.....	24-1
Where the Message Goes.....	24-2
Full Line in <code>/etc/syslog.conf</code>	24-2
The M4 Macro Processor.....	24-4
Logging TCP connections.....	24-5
CHAPTER 25 SYSTEM CRASH ANALYSIS	25-1
The <code>swap</code> Command.....	25-1
The <code>coreadm</code> Command.....	25-4
Examining the <code>/etc/coreadm.conf</code> File in Detail.....	25-4
Patterns Used with the <code>coreadm</code> Command.....	25-5
The <code>adb</code> Command	25-6
The <code>dumpadm</code> Command.....	25-8
CHAPTER 26 VOLUME MANAGEMENT	26-1
Sun Volume Manager.....	26-1
Volumes	26-2
Disk Set.....	26-3
Local Disk Set.....	26-3
Hot Spare	26-3
Hot Spare Pool.....	26-4
Soft Partition.....	26-4
Transactional Volumes.....	26-4
RAID Levels.....	26-4
State Databases.....	26-5
Icons Associated with Volume Management	26-7
Creating State Databases.....	26-8
Using Command Line Tools with State Databases.....	26-12
Understanding the <code>metadb</code> command.....	26-13
Monitoring and Deleting Metadevice State Databases from the SMC	26-16
Working with RAID 0 Volumes	26-17
State Database Best Practices.....	26-34
CHAPTER 27 IPV6	27-1

What Does IPv6 Look Like?	27-2
Understanding IPv6 Within the OSI Model.....	27-2
IPv6 Addressing.....	27-3
Solaris 9 and IPv6	27-3
Interface Auto-Configuration.....	27-3
Configure Solaris for an IPv6 Network	27-5
Key files used by Solaris when connecting to an IPv6 network.....	27-6
Using IPv6 with DNS.....	27-6
Using the ping Command with IPv6	27-6
Using the netstat Command with IPv6.....	27-7
Converting Between Hexadecimal, Binary, and Decimal Numbers.....	27-8
Hexadecimal Numbers.....	27-8
Understanding Binary Numbering.....	27-10
Converting Binary Numbers into Decimal Numbers.....	27-10
Converting between Binary and Hexadecimal.....	27-12
Understanding IPv6 Numerically	27-12
Key Points to Remember	27-14
CHAPTER 28 SYSTEM ACCOUNTING	28-1
Setting up System Accounting.....	28-2
Starting process accounting	28-4
Understanding System Accounting Commands	28-4
The System Accounting Reports.....	28-9
The Daily Report	28-9
The Daily Command Summary	28-14
The Monthly Command Summary.....	28-15
The Last Login Report	28-15
The runacct command.....	28-16
Troubleshooting System Accounting Problems	28-17
How to Repair the tacct file.....	28-18
How to Repair the wttmpx File.....	28-19
CHAPTER 29 DNS.....	29-1
DNS Terms.....	29-1
The DNS Client.....	29-2
The /etc/resolv.conf file.....	29-2
The /etc/nsswitch.conf file	29-2
Understanding nsswitch Template Files	29-5
Files Used with a DNS Server	29-7
DNS Configuration Files	29-9
Types of DNS Resource Records.....	29-10
Using the Apache Web Server with a DNS server.....	29-19
Key Points to Remember	29-20
CHAPTER 30 LDAP	30-1
Why Switch to LDAP	30-2
Sun LDAP Features	30-2
Where Did LDAP Come From?.....	30-3
The Directory Server.....	30-3
Where to Download the LDAP Server.....	30-4
Required DNS Server Setup	30-5
The Directory Console	30-15
The Servers and Applications Tab.....	30-16
The Users and Groups Tab.....	30-17
Working with Users and the Administration Account	30-21
The startconsole Command.....	30-22
Working with the Administration Server.....	30-26

The Configuration Tab	30-28
Files Related to the Administration Server	30-28
Key Files Related to the Administration Server	30-28
The Sun ONE Directory Server	30-32
Directory Basics	30-33
Types of LDAP Directory Servers	30-33
LDAP Terms	30-34
LDAP v3	30-35
Directory Server HTML Documentation	30-35
Introduction to the Directory Server	30-36
The Tasks Tab	30-36
The Configuration Tab	30-37
The Directory Tab	30-40
The Status Tab	30-40
Commands in the /usr/iplanet/ds5/slapd-<hostname> Directory	30-50
LDAP Run Control Scripts	30-54
The LDAP Client	30-58
Other Client-Related LDAP Information	30-59
LDAP Client Access	30-59
Understanding the ldapclient Utility	30-60
LDAP Security	30-60
Understanding the ldapaddent Utility	30-61
LDAP Command Line Utilities	30-62
CHAPTER 31 DHCP	31-1
How DHCP Works	31-2
DHCP Files	31-4
The DHCP Manager	31-5
Configuring a DHCP Client	31-23
Understanding Macros and Options	31-24
Importing and Exporting DHCP Configuration Information	31-28
Command Line Tools to Manage a DHCP Server	31-32
The dhcpconfig Command	31-33
The dhtadm command	31-33
The pntadm Command	31-34
Clearing a DHCP Server	31-36
Client Side DHCP Files	31-38
Client and Server Side DHCP Commands and Daemons	31-38
The /etc/init.d/dhcp Script	31-39

Other Services We Offer

Solaris Consulting

At Avontra Technical Press, we offer a full range of Solaris consulting, including initial system setup, security monitoring and troubleshooting. We can provide on-site or remote secure connections to your servers. One of our most popular programs is the “second pair of eyes” program. In this program, one of our technicians can visit your site to review your current server configuration and security setup. Please contact us at consulting@teachmesun.com for further information.

Solaris Training

Currently, we offer Solaris Certification classes, as well as custom-tailored classes. The author of this book can come to your location . We plan to have scheduled Solaris classes in Denver Colorado; Colorado Springs, Colorado; and Salt Lake City, Utah. Other cities may be included in the future, depending on the demand. We can provide SPARC workstations or can use your facilities. Please contact us at training@teachmesun.com for further information.

Acknowledgments

Ed Winograd served as Technical Editor and Copy Editor for this book. He has twenty-two years of experience in computers and technology, including seven years writing and editing technical manuals, five years teaching Technical Writing at the University level, and many years of writing and delivering computer training. As a Technical Writer, Ed received the award for Best Technical Training Manual in the Rocky Mountain Regional competition of the Society for Technical Communication.

In the past few years, Ed has edited dozens of manuals and many online help files for clients in the networked storage and telecommunications industries. His most recent book edits include a best-selling book on computer game programming, as well as a book on selling and buying antiques and collectibles in flea markets, antique malls, and eBay. His literary edits include a book of multicultural short stories, *Pinch a Lotta Enchiladas*, for which he also wrote one of the stories.

As a Technical Editor, Ed has served on the Style Guide committee for the Storage Products Division of a major worldwide supplier of networked storage systems. He has also edited three newsletters in the computing and high tech fields and has written many articles on computing topics.

Ed received his B.A. in English from Colorado College and a graduate degree in English from the University of California, Berkeley. He is a Senior Member of the Society for Technical Communication, a Songwriter Member of ASCAP, and a Member of the Society of Children's Book Writers and Illustrators.

In his spare time, Ed enjoys storytelling and is webmaster and chief content writer for the website of the Rocky Mountain Storytellers' Conference. He lives in Colorado Springs with his wife Gaynelle and their children, Michael and Julie.

Contact Information:

Email: EdWinograd@ureach.com
Address: 220 Silver Spring Drive
Colorado Springs, CO 80919
Phone: 719-528-5448

Royce Williams participated as a contributing author on this book, he did an excellent job writing the LDAP chapter (Chapter 30) Royce Williams is (among other things) a directory administrator for a medium-sized ISP in Anchorage, Alaska. He's wrangled **sendmail** and LDAP and battled spammers and on behalf of a 50K user base for over three years on systems ranging from Solaris 2.5.1 to Solaris 9 and FreeBSD. He also has a B.A. in CS from the University of Alaska Anchorage on the ten-year plan. In his "spare" time, he collects Alaskan license plates and is an avid movie buff. He occasionally used to wonder why he wasn't an English major until he started reading the writing of Larry Wall. Most things Royce can be found at <http://www.tycho.org>.

Conventions Used in This Book

- **Commands:** In the text, the commands that you type are displayed in a monospace bold font. For example: “The **format** command is used to partition a disk.” All file and directory names are also displayed in a monospace bold font. For example: “the **/etc/passwd** file is used to store user information.”
- **Arguments and Options:** Some commands require arguments and options to pass information to the command itself. Command arguments and options are enclosed in angle brackets `<>`. When you see an argument or option inside the `<>` brackets, type the appropriate value for that argument or option, but not the brackets themselves. For example: `lpr -d <default printer>`.
- **Comments:** Comments are shown in italics. For example: *Always back up this file before editing it.*

Foreword

The Basic Skills section is geared toward novice users. Before using Solaris 9, the reader is introduced to SPARC-based systems. The reader is then guided through a basic installation of Solaris 9, so that the reader's workstation is set up to follow the book's examples. The next couple of chapters cover the Common Desktop Environment, basic file and directory skills, the vi editor, and how to use the different help options available from Sun Microsystems.

The Intermediate Skills section covers typical day-to-day system administration functions. These chapters describe skills such as creating users, performing tape backups and working with software programs. All of these tasks can be performed using only the original Solaris 9 CDRoms and a workstation that is not attached to secondary devices, such as tape drives and modems.

The Advanced Skills section covers how to set up and use the following advanced servers:

DNS (Domain Name Service): Provides hostname to IP address resolution

NIS (Network Information System): Replicates systemwide configuration files

NIS+: An advanced version of NIS

DHCP (Domain Handling Control Protocol): Gives network settings to clients

LDAP (Lightweight Directory Access Protocol): Provides company-wide directory information

Other advanced topics include the Solaris Store Edge Volumes (RAID 0, RAID 1 and RAID 5 volumes) and troubleshooting topics such as analyzing core files and dump files .

One highlight of this book is the LDAP chapter. It features over fifty pages of material and over a dozen examples that illustrate how to set up and use the Sun One Directory Server. Topics in this chapter include how to set up the LDAP server, basic LDAP tree configuration, using LDIF files and Directory Management skills.

Steven Beebe

Chapter 1 Working with SPARC Systems

Lessons in This Chapter

Lesson 1.1 Using the OpenBoot Prompt	1-14
Lesson 1.2 Using OpenBoot Commands	1-18
Lesson 1.3 Using the probe-ide command	1-20
Lesson 1.4 Using the devalias command	1-22
Lesson 1.5 Using the obdiag Command	1-25

Introduction

Solaris 9 is a UNIX operating system produced by Sun Microsystems Incorporated. This operating system will run on both SPARC™ (Scalable Processor Architecture) hardware produced by Sun and Intel™ based systems produced by the Intel™ Corporation. According to Sun Microsystems's press releases, Sun has decided to release the Intel versions of Solaris late in 2003. There was a lot of end user protest and concern about not having an Intel version of Solaris 9 that forced Sun into creating an Intel version of Solaris 9. There is also some unofficial chit-chat on the Internet about having a Linux or Open Source version of Solaris 9, but that's not official news. These facts may change after this book's publication date, because Solaris 9 on Intel is a fluid issue.

Solaris 8 is also a version of Solaris that is produced for both the Intel and the SPARC architectures. With an ever-increasing market share of Linux as a free version of UNIX for Intel-based systems, Sun decided to scale back its development of any type of UNIX operating system that runs on Intel based computers. If a reader wishes to study the Solaris 9 operating system, he or she has one of two possible choices:

The first choice is to use the Intel version of Solaris 8 or Solaris 9 (when it comes out). Over 80% of the hands-on lessons can be performed with the Intel version of Solaris 8 or Solaris 9. Unfortunately, some of the more advanced features of Solaris 9 can not be performed with the Intel versions of Solaris 8 and Solaris 9.

The last Intel version of Solaris 8 is the 02/02 version (February, 2002). The Solaris 8 Media Kit for Intel can be purchased from the online Sun Store located at <http://store.sun.com> for \$45.00 USD. The official Solaris 8 for Intel HCL (Hardware Compatibility List) can be found at:

<http://solde.sun.com/support/drivers/hcl/index.html>

Make sure to check the Solaris 8 HCL against the system that will be used for learning Solaris. The author of this book was rather surprised to find out that Solaris 8 does not support the Voodoo video card in one of his workstations. The HCL lists vary with the date on Solaris 8 media kit. If you have an older version of Solaris 8 for Intel, make sure to match the hardware with the HCL of your version of the Solaris 8 Media Kit.

Sun Microsystems tries to keep each version of Solaris backward compatible with the previous generations. The Solaris operating system and UNIX as a whole tend not to change too drastically between versions. Because of this, the Intel version of Solaris 8 should work with most of the lessons. Companies have too much money invested in software to allow for major changes in UNIX.

The other choice the reader has is to pick up a used SPARC workstation and load Solaris 9 on the workstation. The most affordable way to obtain Solaris 9 is to download the Solaris 9 CDROM images off the

Internet (this option might be discontinued in the future). If the reader can not download the Solaris 9 CDROM images from the Internet, he or she can purchase the Solaris 9 Slim Kit from Sun. The Solaris 9 Slim Kit has a functional but limited selection of CDROMs and a DVD disk. It also comes with a minimum set of paper documentation and manuals that are helpful to have when installing Solaris 9. This kit can be purchased from Sun Microsystems's website <http://store.sun.com>. It will be discussed in greater detail later in the chapter.

A SPARC workstation can be purchased from online auction sites like <http://www.ebay.com>, or <http://www.amazon.com>. The best used SPARC workstation to consider is the Ultra 5 workstation. The best new workstation to consider is the SunBlade 100. The Ultra 5 can be purchased from an online auction site like <http://www.ebay.com> or <http://auctions.yahoo.com>. After Sun Microsystems produced the SunBlade 100, prices on the Ultra 5 workstations dropped like a rock. The SunBlade 100 is a nice workstation because it uses a lot of regular PC components (such as Seagate hard drives, DIMM chips, and standard IDE CDROMs). Most savvy Solaris users will purchase the minimum number of components for the SunBlade 100 from Sun Microsystems and will then purchase additional hardware from third-party vendors.

Real World Experience

- My advice when purchasing a SunBlade 100 is to purchase the smallest configuration possible. Purchase additional hard drives, memory, and DVD drives directly from the manufacturer. Just lift the cover of the SunBlade 100 and see who makes the components. If that does not work, search the Internet message groups and find out what components work on a SunBlade 100.
- Once the one-year hardware warranty from Sun runs out, there really is no need to purchase replacement components from Sun, except for the CPU, motherboard, and a riser board. The following components can be purchased directly from the vendors.
 - ATX power supply (if it becomes damaged : use a standard ATX power supply. The pins are standardized. Check the power supply's dimensions so that it will fit in the case.)
 - Seagate hard drives or Western Digital hard drives will work.
 - Standard 3-1/2" floppy drive.
 - LiteOn DVD drive (don't purchase a CDROM drive—, CDROMs are on their way out.
 - DIMM chips with ECC (Internet purchases are the only place to find these. The best supplier is <http://www.crucial.com>.)

Steven Beebe
Author

How to Purchase an Ultra 5 Workstation

The Ultra 5 workstation is compatible with Solaris 9. Figure 1.1 is a photograph of an Ultra 5 workstation.

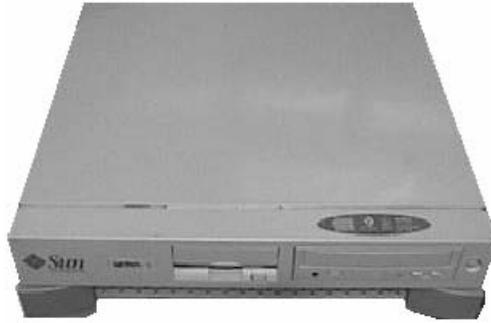


Figure 1.1 Ultra 5 Workstation

For readers on a budget, purchase the Ultra 5 on the Internet, use it for studying Solaris 9, and then resell it for the same amount of money as the original purchase. Think of this more as a rental than a real purchase. Try to find an Ultra 5 with the following components:

- 96 MBRAM as a minimum, 256 MB are needed for the LDAP server.
- A minimum 4 GB system disk. The system only needs one hard drive. However, the lessons in Chapter 27 (Volume Management) and Chapter 13 (Hard Drives) require the use of a second hard drive. Just purchase a really cheap 500 MB hard drive for these lessons. The second drive just needs to be functional—the size does not matter.
- A Sun keyboard and mouse. Some lessons in this book use the STOP button that is found only on Sun Keyboards.
- There are two versions of the Ultra 5. One version has an SVGA video card, and the other has a 13W3 video card. The SVGA version is more popular and more expensive. The 13W3 version is less popular but cheaper.

Try to find a system that has a rather slow CPU. None of these lessons really puts a strain on the CPU. When it comes to memory, try to purchase a system with 256 MB of memory. buying anything more than 256 MB will just be wasted money. The Ultra 5 workstations take a proprietary version of memory, which is rather expensive. The minimum amount of memory needed by Solaris 9 is only 96 MB. This setup should cost about \$250.00.

Some Ultra 5 workstations come with a 13W3 video port and others come with a standard SVGA video port. If the reader purchases an Ultra 5 with a 13W3 video port, either a Sun monitor or a 13W3 to SVGA adapter is also required. The Sun monitor is a great quality monitor, but it tends to be rather expensive. Also, understand that the monitor does not have a standard SVGA adapter, so it can only be used with the Ultra 5. The other choice the reader has is to purchase a 13W3 to SVGA (13W3 Female to HD-15 Male) adapter. This adapter will let the reader use a standard SVGA monitor with the 13W3 version of the Ultra 5. Figure 1.2 shows this adapter.



Figure 1.2 13W3 to SVGA Adapter

This adapter should only cost about \$7.00, in addition to a \$5.00 shipping fee. With this adapter, a standard SVGA monitor can be used to view the output of the Ultra 5. A standard SVGA monitor is a lot cheaper than a Sun monitor. This adapter should deliver at least 800x600 24-bit color, according to the official specifications. The best place to find this is on <http://www.ebay.com>. Type in a keyword search as “13W3 SVGA.” Remember, it is a 13W3 MALE to SVGA FEMALE. There are different types of 13W3 adapters, so check the adapter very carefully before purchasing.

Try to avoid purchasing small items from what are known as Sun VARs (Value Added Resellers). A Sun VAR is a company that has formed a partnership with Sun to sell Sun equipment. Sun VARs are valuable for small businesses that need to purchase entire systems and need a qualified system integrator to set up the equipment. Unfortunately, Sun VAR prices for small items are almost always higher than a direct purchase from Sun.

How to Purchase a SunBlade 100 Workstation

For readers that want a new workstation with a one-year warranty, the SunBlade 100 is the best system to consider. The Solaris 9 O.S. has a minimum system requirement of 96 MB of RAM and 2.9 GB of hard disk space. The SunBlade 100 is a Sun workstation that currently sells for less than \$1000 new. The cheapest place to purchase a SunBlade 100 is from the online Sun Store at <http://store.sun.com>. The SunBlade 100 is a very good workstation to learn Solaris 9 because it has a standard SVGA video card and IDE disk drives. This workstation can be attached to a standard SVGA monitor and delivers 1024 x 768 24-bit color. Because this is a new workstation, upgrade parts can still be purchased directly from Sun or from third-party manufacturers. The SunBlade 100 uses IDE disk drives, so an extra hard drive can be added to the system at a cheap price, which helps keep down costs. The SunBlade 100 is SPARC compatible, so it can be used with Solaris 8 (SPARC edition.) but not with earlier versions of Solaris. Solaris 7 and previous versions of Solaris do not have the drivers to work with the SunBlade 100. On the bright side, Sun will support the SunBlade 100 through several new generations of Solaris.

Solaris 9 Media Kits

Sun Microsystems distributes the Solaris 9 operating system through media kits and through the Internet. Remember, Solaris 9 has been developed for the SPARC architecture only. There are no Intel CDROMs or Solaris 9 Intel media kits available.

The Solaris 9 Media Kit includes the Solaris 9 CDROMs, a DVD disk, and some paper documentation (electronic versions of the documents are online). The best place to buy these media kits is directly from Sun. The media kits have a 4-digit release date in the format Month/Year. For example, a Solaris 9 05/02 Media Kit was produced in May, 2002. This indicates the last month that Sun burned the CDROMs for that media kit.

It is important to check the date when the media kit was produced because new media kits include extra software and patches that older media kits might not have. For example, the Solaris 8 01/00 (January, 2000) Media Kit would not have the same software as the Solaris 8 02/02 (February, 2002) Media Kit. The same rules will apply to Solaris 9 Media Kits when new versions are created.

The media kit comes in a cardboard box that contains CDROMs, DVDs, and some documentation booklets. This box is shipped via FedEx within twelve days of the purchase order (rush orders are available). There are two different versions of the media kit, the System Administrator's Kit (w/English Hardcopy Documentation) or the Slim Kit. Figure 1.3 is a picture of the Slim Kit.

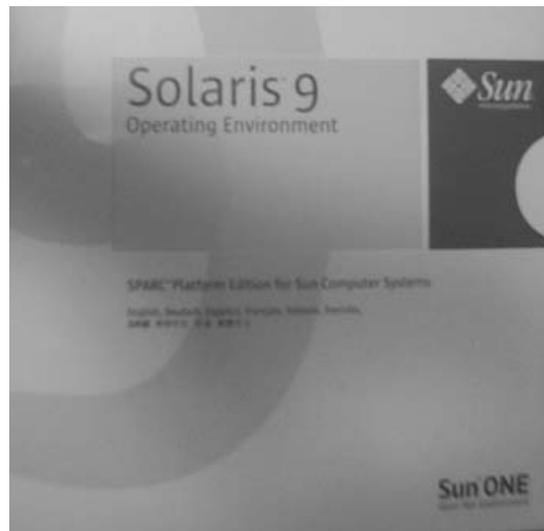


Figure 1.3 Solaris 9 Media Kit

To purchase either the Slim Kit or the System Administrator's Kit, either:

- Visit Sun's website at <http://store.sun.com>, or
- Call the Sun Store U.S. at 1-800-Sun-0404, Option 6.

Purchases made from Sun's website are cheaper than purchases made on the phone. Sun is trying to encourage users to make online purchases by giving these price discounts.

The online version of the Sun Store requires a registration. The registration asks the buyer for a shipping address, telephone number, and a major credit card (if a purchase will be made). Most of the time, the online Sun Store will offer discounts for using the Sun Store webpage.

Sun Microsystems also distributes the Solaris 9 operating system through the Internet in a program known as the "Binary License Program." This program lets a company or individual download CDROM images. These images can be used by CD burning software like Adaptec Easy CD Creator™. The only drawback is that CDROM images downloaded from the Internet can have flaws, due to the download process. If possible, the author purchases CDROMs, instead of downloading the CDROM images off the Internet. The operating system is too critical to risk having damaged files from a large file download off of the Internet.

The Solaris 9 Beta Refresh version of Solaris has the CDROM images broken down into zip files. That is not too bad, except that for some strange reason, these zip files require the execution of a Bourne shell script to be unpacked. That's not a problem if an extra UNIX server is available with four or five GB of free disk space. If a purchaser does not have a UNIX server available, creating the CDROM images is a problem. How do you un-

compact Solaris files that require a Solaris server? OOOOPSS. . . One possible way to un-compact the CDROM images is to install Linux on an Intel-based computer and run the Bourne shell script from there.

Solaris 9 Slim Kit

The author of this book highly recommends the reader purchase the Solaris 9 Slim Kit. It has all the software and CDROMs needed for this book. Other kits cost more and have advanced servers and software that a typical administrator might not need. This kit is for system administrators that just want the basic Solaris 9 operating system. This price and content quoted here are based on the July, 2002 prices. Check Sun Microsystem's official website for the latest prices and contents. This kit has a list price of \$50.00 USD.

The Slim Kit includes the following items:

- Solaris 9 Installation CD for Sun Hardware
- Solaris 9 Operating Environment Software CDs (2)
- Solaris 9 Online Documentation CDs (2)
- Solaris 9 Languages CD

Solaris supports the following languages:

- French
- German
- Italian
- Japanese
- Korean
- Spanish
- Swedish
- Simplified Chinese
- Traditional Chinese.

- Solaris 9 Supplement CD

Contains the following optional software components:

- SunATM™ 5.1
- Sun Enterprise™ 10000 Capacity On Demand 1.0
- SunFDDI™/P 3.0
- SunFDDI/S 7.0
- SunHSI/P™ 3.0
- SunHSI/S 3.0
- Netra™ ct Platform 1.0
- Lights Out Management 2.0
- Netra™ t11xx Alarms 2.0
- OpenGL™ 1.2.2
- PC launcher 1.0.1
- PC file viewer 1.0.1
- Sun™ Remote System Control 2.2.1
- ShowMe TV™ 1.3
- Sun Enterprise™ 10000 SSP 3.5
- Sun Fire™ 15K SMS 1.2
- SunForum™ 3.1
- SunVTST™ 5.0

Java 3D™ 1.2.1_04
Solaris on Sun™ Hardware Documentation

- DVD Install Disk - also included in the Solaris 02/02 Media Kit is a DVD that includes the contents of all the previously listed CDROMs.

Solaris Software Companion CD - freeware and shareware software that is unsupported by Sun Microsystems.

Three booklets that are included with the Slim Kit :

Start Here/Installation Instructions for Solaris 9 software
Solaris 9 Release Notes
Solaris License Information

Solaris 9 System Administrator's Kit

The System Administrator's Kit (w/English Hardcopy Documentation) includes all the CDROM and DVD installation disks and some additional software from Oracle, Netscape and other third-party software packages for the Solaris 9 Operating Environment (Operating Environment is the same term as Operating System). It also includes some extra booklets that cover various install topics. The System Administrator's Kit includes all the contents of the Slim Kit and has the following extra components:

Booklets

- Solaris 9 Installation RoadMap
- Solaris 9 Installation Guide
- About Solaris 9 Document
- Solaris 9 What's New Document
- Solaris 9 Hardware Platform Guide

Bonus Software CDROMs:

- Oracle9i Database (3 CDs)
- StarOffice™ 6.0
- StarSuite™ 6.0
- Sun ONE Developer Studio for Java™ Community Edition 3.0 (formerly Forte™ For Java 3.0, Community Edition)
- Forte Developer 6, Update 2
- GNOME 2.0 Desktop

Sun ONE Advantage Software:

- Sun™ ONE Integration Server EAI Edition 3.0 (formerly iPlanet™ Integration Server EAI Edition 3.0)
- Sun ONE Message Queue for Java 2.0 (formerly iPlanet Message Queue for Java)
- Sun ONE Application Server, Enterprise Edition 6.5 (formerly iPlanet Application Server, Enterprise Edition 6.5)

Sun changes its media kits from time to time, so this information could be out of date when the book is published. Sun Microsystems is an aggressive company that adds the latest features as fast as possible.

Early Access Solaris 9

There are some media kits and CDRoms labeled “Early Access” on the cover. The Early Access version of Solaris 9 is basically the beta version. This version can be used for an individual or company that just wants to *play around* with Solaris 9, but it is not suitable for software development or use in a production environment.

Sun Microsystems changes its free download policies from time to time. When this chapter was first written, the Early Access version of the Solaris 9 Operating Environment was available for download from the Internet for free. This policy might be different by the time this book comes into production. The Early Access program will be discontinued at some time in the near future.

The Early Access CDRoms images are:

sol-9-install-br-sparc-a.zip	Installation CD, Segment 1 of 3	98.68 MB
sol-9-install-br-sparc-b.zip	Installation CD, Segment 2 of 3	107.95 MB
sol-9-install-br-sparc-c.zip	Installation CD, Segment 3 of 3	125.50 MB
sol-9-br-sparc-vi-a.zip	Software 1 of 2 CD, Segment 1 of 3	110.38 MB
sol-9-br-sparc-vi-b.zip	Software 1 of 2 CD, Segment 2 of 3	125.52 MB
sol-9-br-sparc-vi-c.zip	Software 1 of 2 CD, Segment 3 of 3	91.59 MB
sol-9-br-sparc-v2.zip	Software 2 of 2 CD, Segment 1 of 1	215.90 MB
sol-9-doc-br-v1.zip	Documentation CD, Segment 1 of 2	194.45 MB
sol-9-doc-br-v2.zip	Documentation CD, Segment 2 of 2	220.95 MB
sol-9-lang-br-sparc.zip	Languages CD	311.80 MB

Optional Download Components Include :

Script for the S9 CD Images
Labels for first two CDs
Labels for second pair of CDs
Labels for first two CDs (to print on A4)
Labels for the second pair of CDs (to print on A4)
README - Online Documentation

For more information, see the Solaris 9 Customer Early Access CD Images official website

<http://www.sun.com/solaris/programs/solaris9ea>

As mentioned earlier, the Early Access program will be discontinued at some time in the near future. Check the <http://www.sun.com> website for a possible free binary program. If Sun does not have free CDRom images on the Internet, purchase the Solaris 9 Slim Kit.

Licensing Solaris 9

The Solaris 9 binaries can be downloaded for a fee, or media kits can be purchased from Sun. These fees do not cover the license to use the Solaris 9 Operating System. Purchasing the media and purchasing a license to use the Solaris 9 operating environment are two different issues. Sun’s software licenses are based on the number of CPU’s that are actively running in a server. Also, understand that some software that comes with the Solaris 9 Media Kit requires a separate license. The Sun One Directory server is an example of this. The Directory Server is only free for a 60-day trial period, then it needs to be purchased.

For systems with only one processor :

- All Sun SPARC hardware with one CPU that is purchased directly from Sun comes with a license to use the current Solaris Operating Environment, SPARC Platform Edition, for free. This includes new SPARC

systems, as well as remanufactured SPARC systems from Sun. The license includes any current or previous version of the Solaris operating system. If a company already has the Solaris 9 CDRoms, they can be used for the installation of Solaris 9. In a nutshell, if a system was purchased from Sun, the current operating system can be installed for free.

- If a used system with one CPU is purchased from a third-party seller, that system does not have a license to use the Solaris Operating Environment, SPARC platform. To use Solaris 9, or any version of Solaris, on a used system, it is necessary to purchase a media kit from Sun, or to download the Solaris 9 binaries from Sun. After the Media Kit is purchased or downloaded from the Internet, the Solaris 9 operating system must be registered with Sun. The purchase price of the download or the Media Kit is the license fee.

For systems with more than one processor, a company needs to purchase the Solaris 8 Media kit. Then an appropriate Solaris 8 license must be purchased for the server. After the Solaris 8 Media kit is purchased and the license is purchased, the Solaris 9 upgrade license must be purchased. Table 1.1 shows the license fees (this might change in the future, so check with Sun for official prices).

Upgrade Version	Number of CPUs	Price
Solaris 9 Workgroup Server 2 Upgrade	2 CPU	\$249
Solaris 9 Workgroup Server 4 Upgrade	4 CPU	\$999
Solaris 9 Midrange Server 9 Upgrade	8 CPU	\$6,000
Solaris 9 Midrange Server 16 Upgrade	16 CPU	\$20,000
Solaris 9 Midrange Server 32 Upgrade	32 CPU	\$60,000
Solaris 9 DataCenter 64 Upgrade	64 CPU	\$160,000
Solaris 9 DataCenter 128 Upgrade	128 CPU	\$400,000

Table 1.1 Solaris 9 License Fees

Quick Tip

- Because the Solaris 9 license is free for systems with only one CPU, it is cost beneficial to purchase a system with only one CPU, if possible. Once a system has more than one CPU, the costs rise dramatically. Try not to exceed four CPUs. The fifth CPU will make the cost jump from \$1,000 to \$6,000 dollars.
- Purchase the Solaris 9 Slim Kit, instead of downloading the CDROM images off the Internet. When it comes to downloading CDROM images, there is the of corruption in the CDROM ISO image or in a file. When it comes to something as critical as an operating system, the original CDROM images ensure that file corruption issues are not a problem.
- Purchase the Solaris 9 System Administrator's Kit only if there is a specific software package included with the Media Kit that is needed with a server. The documentation can be found on <http://docs.sun.com>, so a hard copy is not

How to Use a SPARC System

Most computer technicians that work with Intel-based computers are familiar with the BIOS (Basic Input Output Software) that configures the motherboard of the system. This hard coded software resides on the motherboard. The BIOS chip performs basic hardware diagnostics before the operating system is loaded. After running the motherboard diagnostics, the BIOS software boots an operating system like Microsoft Windows 98 or Microsoft Windows XP. On most Intel-based systems the F1 key or the key combination CTL + ALT + ESC is used to access a BIOS screen. This screen lets a user change the BIOS options on the motherboard, such as the BIOS date, hard drive parameters, and power management options.

This section is very difficult to understand. Please, re-read the remainder of this section several times to understand the OpenBoot firmware. Also, understand that a PC BIOS is only slightly similar to OpenBoot firmware. Drill this into your head:

OpenBoot Acts Different Than a BIOS

Sun Microsystems produces its own type of hardware, known as SPARC hardware. This hardware is incompatible with most Intel devices or Intel chips. The SPARC hardware is only officially produced by Sun Microsystems. SPARC hardware uses what is known as the OpenBoot firmware. The OpenBoot firmware is also known as the monitor program. In some limited ways, the OpenBoot firmware functions like an Intel motherboard's BIOS.

The biggest difference is that the OpenBoot firmware and a BIOS is that OpenBoot will let the system administrator *see* the operating system when it is running. Another difference is that the OpenBoot screen can be seen through either a monitor or through serial port A. The BIOS screen can only be seen on a monitor when the operating system is not running.

When a SPARC-based system first starts (that does not have an operating system installed) the OpenBoot firmware displays a screen interface known as the "monitor program." The monitor program is also known as the OpenBoot PROM (Programmable Read-Only Memory). When the OpenBoot monitor program is running, an **OK** prompt will appear on the screen. This is where a system administrator can change the firmware settings on the motherboard. Once an operating system has been installed on the Sun workstation, the operating system is seen through the OpenBoot firmware. If the system administrator types some reserved keyboard strokes, the original **OK** prompt can be seen and used. Have patience trying to understand this—it may be a foreign concept if you're used to Intel-based systems.

There are two methods of accessing the OpenBoot prompt(the **OK** prompt.)

- 1) On the back of every SPARC computer there is a port labeled "Serial Port A."
One of three different devices can be attached to Serial Port A.

- A null modem cable. The other end of the cable is attached to another Sun workstation or a Microsoft Windows computer or any other device that can type text into a null modem cable.
- A terminal. Understand that a terminal can only display text, it is not much more than a monitor and keyboard.
- A Terminal Server. A Terminal Server is a device that has a null modem cable on one side and an Ethernet card on the other side. This lets the system administrator access the serial port through a router.

- 2) If a monitor, keyboard and mouse are attached to a SPARC system and the system has a video frame buffer, the OpenBoot prompt can be seen through the monitor. The OpenBoot PROM still presents a command line interface only—there is no graphical interface to the OpenBoot PROM.

The OpenBoot PROM performs the same types of functions as the BIOS screen found on Intel-based computers. There are some key differences, however, such as:

- SPARC-based systems have two motherboard chips, the PROM chip and the NVRAM chip. Intel-based motherboards usually have only one BIOS chip.
- Serial Port A can be used to access the OpenBoot PROM on SPARC hardware. Intel-based computers can only display a screen through a directly connected monitor.
- On Intel-based systems, the BIOS screen is only accessed before the operating system starts. Once an operating system like Windows XP starts, the BIOS settings can not be accessed or changed. The variables saved in the OpenBoot PROM can be changed from within the operating system. It is impossible to change the motherboard settings while running any version of Windows.
- The OpenBoot PROM has diagnostics utilities that can run extensive hardware tests on internal devices and devices attached to the outside of the server.
- The OpenBoot PROM can read device drivers from the chips on the devices and pass the device driver information on to the operating system. For example, if a company purchased a new Ethernet card for a server, the device driver might be located on the card itself (depending on the manufacturer).

Sun Keyboards

Sun Microsystems produces a special keyboard that works with SPARC hardware. Figure 1.4 shows a picture of a Sun keyboard, and Figure 1.5 shows an enlargement of the STOP key.



Figure 1.4 A Sun keyboard

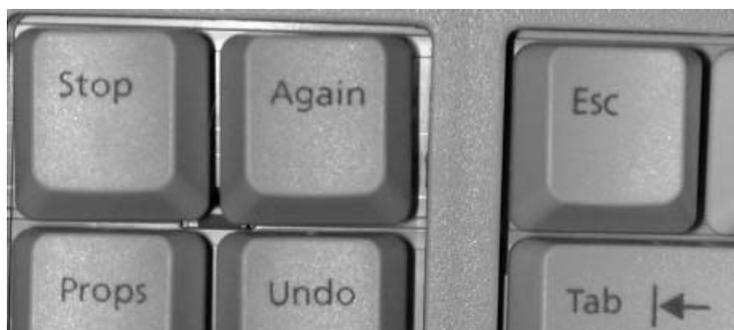


Figure 1.5 An Enlargement of the STOP key.

There is a STOP key located on the upper right corner of the keyboard. The STOP key can be used in combination with other keys for the following results:

- STOP + A Stop the operating system and display the **OK** prompt.
- STOP + D Reboot the server and perform extensive diagnostics.
- STOP + N Reboot the server and reset the PROM values to the default values (from the PROM chip .)

The OpenBoot Prompt

If a monitor is connected to a video card on a SPARC system and the system is turned on, some diagnostic messages about testing memory and system devices are displayed. If a video card is not present, a terminal connected to Serial Port A will also display the same system test messages. If a Sun keyboard is attached to the system, the simultaneous keystroke combination of STOP + A will stop the operating system from starting. If a BREAK signal is sent from a terminal to the server, the operating system will also stop, and the system will halt to the **OK** prompt (the OpenBoot program). The **OK** prompt will also be displayed if the operating system crashes or the operating system is brought down to run level 0. If the PROM variable **auto-boot?** is set to false, the operating system (Solaris 9) will not automatically start.

There were four different versions of the OpenBoot PROM created over the years. They are:

- OBP 1 – The very first OpenBoot prompt. OBP 1 was used with the original Sun SPARCstation 1.
- OpenBoot PROM version 2 – used with the Sun SPARCstation 2.
- OpenBoot PROM version 3 – used with Ultra series computers.
- OpenBoot PROM version 4 – used with the SunBlade 100 computers and some 2001 year systems.

Some very old workstations, like the Sunstation 5 and Sunstation 2 workstations, use the OpenBoot PROM version 2. These very old systems are generally incompatible with Solaris 9 because they don't have the right hardware. The most common OpenBoot versions used with Solaris 9 are going to be OpenBoot PROM version 3 and OpenBoot PROM version 4.

The OpenBoot PROM 3 and 4 interfaces use the Forth programming language. A typical system administrator does not need to understand the Forth language. Only extremely high end software developers and network consultants might use the Forth language to create custom-made Forth scripts. Just understand that the OpenBoot PROM is based on the Forth language.

Another key feature of OpenBoot PROM versions 3 and 4 is that they are the only versions of OpenBoot that can be "flashed," which means that you can upgrade the motherboard chips from a file downloaded off the Internet. These OpenBoot flash files are downloaded from one of Sun's official webpages.

Most SPARC workstations only have one motherboard, while most high end SPARC servers have multiple motherboards. The "clock board" is the most important one—it controls all the others.

Systems with only one motherboard are :

SPARCstation 4, 5, 10, 20
Ultra 1, 2, 5, 10, 30, 60, 80, 220, 240, 420, and 450

Systems with multiple motherboards are :

Enterprise 3500, 4500, 5500 and 6500 Series. Also, the 3x00, 4x00, 5x00 and 6x00 series

Sun hardware changes all the time. The lists above only cover the most common types of Solaris servers. For systems with multiple motherboards, there are some special considerations:

- There is a backplane board that allows the different motherboards to communicate with each other. The clock board (one of the motherboards that controls system timing) controls the backplane communications.
- The clock board has the official HOSTID and the official Ethernet address. The other lower ranking motherboards inherit the HOSTID and Ethernet address from the clock board.
- At startup, each CPU and PROM is checked
- At startup, each CPU is diagnosed separately with its own Power On Self Test (POST).

When a server is turned on, there are certain tasks that have to be performed by the OpenBoot PROM. These tasks include :

1. Determine the system's configuration.
2. Test and initialize the hardware connected to the system.
3. Load any FCode device drivers that are hardware-burned into devices.
4. Create a device tree (the operating system reads this and configures the kernel to operate with the devices attached to the server. It adds additional device drivers inside the operating system, if necessary).
5. Boot the operating system from a local device (hard drive) or over a network device (ethernet card).

Using a Null Modem Cable

To access the OpenBoot prompt, a null modem cable can be used between a regular PC and the SPARC based system. Figure 1.6 is a picture of a null modem cable.



Figure 1.6 Null Modem Cable

The large end of the cable is a male DB-25 and the small connector on the cable is a female DB-9 connector. The large connector is attached to Serial port A on the SPARC-based system. Figure 1.7 shows the DB-25 attached to a SPARC workstation. (just as a side note, notice in Figure 1.7 the 13W3 to SVGA Adapter attached to the SPARC workstation.)



Figure 1.7 Null Modem Cable Attached to a SPARC System

The small connector is attached to a standard COM1 port on a PC or laptop. The top image in Figure 1.8 shows the end of the cable. The bottom image in Figure 1.8 shows the serial port on a laptop, that the cable connects to.

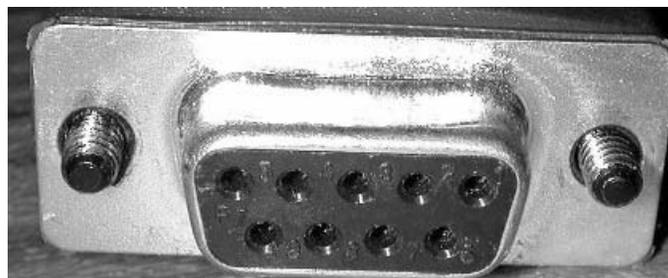




Figure 1.8 Null Modem Cable Attached to a Laptop

Once the null modem cable is attached, open the HyperTerminal program in Microsoft Windows. To run HyperTerminal, click Start > Programs > Accessories > Communications > HyperTerminal

Quick Tip

- If HyperTerminal was not installed with Microsoft Windows, follow these instructions:
 1. Click on Start > Settings > Control Panel>.
 2. Double click on the Add/Remove Programs Icon.
 3. Left click on the Windows Setup tab at the top center of the window.
 4. Left click on Communications.
 5. Left click on the Details... button.

Add the HyperTerminal application to Windows

Configure HyperTerminal to have the following properties:

Communications : COM1 or COM2 (where the null modem cable is attached to the PC.)
Emulation : VT100
Bits per second : 9600
Data bits : 8
Parity : None
Stop Bits : 1
Flow Control : Hardware

Lesson 1.1 Using the OpenBoot Prompt

This lesson can only be used by readers that have purchased or obtained a SPARC system. This lesson is primarily designed for Sun computers that have a monitor and keyboard directly attached to the system. A null

modem cable can be attached to Serial Port A to view the OpenBoot prompt. For readers without a SPARC system, it is a good idea to read through this lesson for a better understanding of how the OpenBoot prompt works, even if a Sun workstation is not available for the lesson. For readers with the Intel version of Solaris 8 and an Intel-based computer, understand that Intel-based computers do not have an OpenBoot prompt. Instead, they use what is known as a BIOS screen for hardware configuration.

1. Power on a workstation (this workstation must have a monitor and keyboard attached directly).
2. Hold down both the STOP + A keys immediately as the workstation starts to boot.
Figure 1.5 is a picture of the STOP key. The STOP key is located on the top left side of the keyboard.
3. For systems that have a null modem cable attached to the SPARC workstation, configure HyperTerminal as described in the previous section. Start a HyperTerminal session and then press CTL + Break on the PC keyboard.
An OK prompt should be visible at the top of the screen.
4. Type the command **setenv auto-boot? false**
*When the **auto-boot?** variable is **false** the system will not automatically boot the operating system but will instead halt at the OK prompt.*
5. Type the command **banner**
*The command **banner** shows various pieces of system information, such as the system's memory, Ethernet address, and Host ID.*
6. Type the command **power-off**
*The command **power-off** turns off the power on most SPARC workstations and servers.*
7. Power on the system
8. Type the command **printenv**
*The **printenv** command shows the current PROM variables. Press the Return key to see the next line of text. If the space bar is pressed it will display an entire screen full of text at a time.*
9. Type the command **setenv auto-boot? false**
*Just as a backup, the **auto-boot?** variable is set to false again. When the **auto-boot?** variable is **false** the system will not automatically boot the operating system but will instead halt at the OK prompt.*
10. Type the command **printenv auto-boot?**
*When **printenv** is given a variable, it will show only the value of that variable.*
11. Type the command **power-off**
*The system should now power down. If it doesn't, manually power down the system.
Type the command **boot disk** at the OK prompt to boot the operating system. For now, keep the **auto-boot?** variable to false so the operating system does not automatically start.*

The PROM Chip

The PROM chip is responsible for the initial system tests and the booting of the operating system. It has an 8 KB memory and is located on the motherboard. The PROM firmware resides on this chip and produces an OK prompt when the system starts, provided the **auto-boot?** variable is set to false. The PROM chip contains the following items:

- User Interface Commands – The OK prompt, **printenv**, **setenv**, and other diagnostic/interface commands to control the behavior of the firmware
- POST – Power On Self Test.
- Generic Device Drivers
- Default Parameters – Set by the factory, can be reset if the PROM becomes incorrectly configured

The NVRAM Chip

The NVRAM chip is another critical chip on the motherboard. This chip can be identified by a yellow bar-code sticker. It has a very important security number known as the HostID. The HostID is used by software programs to create and maintain licenses. If for some reason the SPARC workstation's motherboard dies, make sure to pull out the NVRAM chip.

This chip can be transferred to another motherboard. If the chip goes bad, Sun Microsystems can produce another chip with the same HostID, but the original chip will have to go through a verification process to make sure that a hacker does not order an NVRAM chip with the different HostID to create a *spoof server*. Figure 1.9 is a picture of an NVRAM chip. Remember, the NVRAM chip has a yellow bar code on the top.

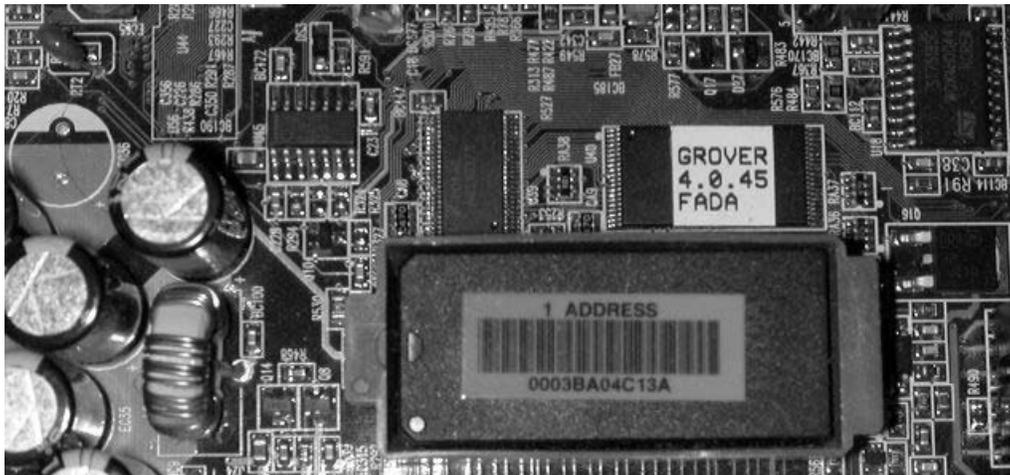


Figure 1.9 NVRAM Chip on a Motherboard

The NVRAM chip contains the following items.

- User-defined custom parameters (such as device aliases)
- HostID – an identification number used with software licenses
- Ethernet address – also known as the MAC Address
- Time of day clock – the system's motherboard clock
- Lithium battery - used to keep the system variables and clock settings when the system is powered down

Working with Sun SPARC Equipment

There are two ways to gain access to the **OK** prompt. Both are covered below.

Situation 1. The system not powered on – Turn on the power and press the STOP + A keys simultaneously. If the **auto-boot?** variable is set to **false** the system will not load the Solaris 9 operating system and will show the **OK** prompt by default. The operating system can then be started by running the **boot** command. The **boot** command supports the following options :

```
boot < options > < device >
```

```
< options >
```

- s** boot into single user mode, used for system maintenance.
 - d** boot into diagnostics mode, perform extensive verbose diagnostics.
 - r** reconfiguration boot, used when a new device is attached to the system. The **-r** option prompts the OpenBoot firmware to perform very detailed hardware scans.
 - V** used with other options **-Vs**, **-Vd**, **-Vr** to display verbose information with the single user boot; diagnostics start and reconfiguration start.
- boot -Vs** will boot the default device specified in the **boot-device** variable in single user mode (because of the “s” option) and with extra information (because of the **-V** option).

< devices >

The device can be any of the following devices that support a Solaris 9 boot.

- disk – a hard drive.
- cdrom – the first CDROM can be booted as a functional copy of Solaris 8.
- network – a network card that support diskless workstations or network boots.
- floppy – for Intel-based systems, a floppy disk can be bootable. This boots automatically if the BIOS is set to boot from a floppy before the C drive.

For example :

boot	Boots the default device listed under the boot-device variable.
boot disk	Boots the hard drive that the alias disk refers to.
boot cdrom	Boots the CDROM (provided the CDROM has a bootable partition)
boot network	Boots from a network card if the network card supports such a boot and if there is a boot server on the same network segment.

Situation 2. If a system is already on, type the **init 0** command at the keyboard or through a terminal connected to Serial Port A. If a monitor is attached to the system, type the command **init 0** through a terminal window or a command line.

Lesson 1.2 Using OpenBoot Commands

In this lesson the reader is going to practice various OpenBoot commands. Because an operating system has not been installed on the system, some of the OpenBoot variables and commands will have to be demonstrated after the operating system has been installed.

1. Power on the workstation.
2. Press the **STOP** and **A** key if the operating system begins to load.
*If the OpenBoot variable **auto-boot?** is set to false, the operating system will not load. If the **OpenBoot** variable is set to true, the operating system will load after the system hardware diagnostics.*
3. Type the command **printenv**
*The **printenv** command shows all the OpenBoot PROM variables. The space bar scrolls down one page at a time and the ENTER key scrolls down one line at a time. If a variable is specified after the **printenv** command, only that variable is shown. The **printenv** command can not change the value of a variable.*
4. Press the space bar until the **OK** prompt returns.
5. Type the command **printenv**
6. Press the Return key until the **OK** prompt returns.
7. Type the command **printenv output-device**

8. Type the command **reset-all**
*The command **reset-all** is used to reset the system just as if the power was turned off and then back on.*
9. Type the command **devalias**
This command shows all the pre-set aliases set up by the factory.
10. Type the command **help**
*This command shows a basic help menu. Look through the help menu. Only type the first word of the topic. For example, to look up the help screen for the “Memory access” choice, type **help memory**. Memory is the first word in the help topic “Memory access.”*
11. Type the command **help power**
This command gives detailed information on the power options
12. Power off the system, type the command **power-off**.

Some Other Useful PROM Command

help	This command displays instructions on the help system. help <category> - displays help for commands in a category help < command > - displays help text for a specific command
printenv	This command shows the NVRAM variables.
setenv	This command sets the NVRAM variables.
reset-all	This command resets the system (acts like the power switch was cycled off and on .)
set-defaults	This command resets all the NVRAM settings back to the original values. The original factory values are stored in the PROM chip. Two modes of operation are: <ul style="list-style-type: none"> • set-defaults - resets all NVRAM variables back to their original default values. • set-default <NVRAM Variable > - resets a specific value back to the default value.

WARNING

If the command **set-defaults** is typed, the **auto-boot?** variable is set to true, so the system will try to automatically boot the operating system when the power is turned on. If the reader is still trying to learn the OpenBoot system, press the STOP + A keys when the system starts again, and reset the **auto-boot?** variable to false with the command **setenv auto-boot? false**.

devalias	Used to create custom aliases;this command also shows all the device aliases on a system.
devalias alias	Shows the device node associated with an alias. Devalias <alias name> <device path> –creates a new alias to represent a hardware device.

show-devs Shows all devices recognized by the system; very good for hardware troubleshooting.

banner Displays information about the server's hardware. The banner command shows the following pieces of information, as shown in Figure 1.10.

**Sun Blade 100 (UltraSPARC-IIe), Keyboard Present
OpenBoot 4.0, 128 MB memory installed, Serial # 50643816
Ethernet address 0:3:ba:4:c4:3d, Host ID: 8034d13b**

Figure 1.10 The Banner Command Output

The **banner** command displays the following output:

Model	The system's model name, including type of CPU
Keyboard	If a keyboard is present
OpenBoot Version	System OpenBoot version
Memory	Detected RAM
Serial #	The serial number
Ethernet Address	The Ethernet address on the NVRAM chip
HostID	A special number used for software licenses

probe-ide Scans the system for all IDE (Integrated Device Electronics) devices

probe-scsi Scans the system for all SCSI (Small Computer System Interface) devices

probe-scsi-all Scans all SCSI devices, including internal and external devices.

probe-fcal Probes all Fibre Channel interfaces.

power-off Shuts off the power to the system.

Lesson 1.3 Using the **probe-ide** command

This lesson assumes that the reader has an Ultra 5 or a SunBlade 100. For systems with SCSI disks, type the command **probe-scsi** instead of **probe-ide**. This command should show information about the hard drive or CDROM, such as Cylinders, Heads, and Sectors. If a hard drive is present but it can not be seen with the **probe-ide** or **probe-scsi** command, it will not be seen inside the Solaris operating system. Check the hardware and cables. Type the command **boot -r** to notify the operating system that a new device was added to the system.

1. Power on the system.
2. Press the STOP and A keys simultaneously.
3. At the OK prompt, type the command **setenv auto-boot? false**
4. At the OK prompt, type the command **reset-all**
5. At the OK prompt, type the command **probe-ide**
6. If no IDE devices are on the system, type the command **probe-scsi**

**USE THE FOLLOWING COMMAND ONLY IF A NEW DEVICE IS ATTACHED TO THE SYSTEM
!**

1. If a new hard drive was added to the system, type the command **boot -r**

The **boot -r** command tells the Solaris operating system that a new device has been installed, so that the operating system will load new device drivers into the kernel.

NVRAM Variables

These variables affect the way the server behaves before an operating system is running. Table 1.2 shows the default NVRAM variables and how they affect the operation of SPARC hardware. Default values are shown in parentheses.

Variable	Values (Default)/alternate	What these values do
auto-boot?	(true)/false	<ul style="list-style-type: none"> • True – boots the operating system automatically after the system is shut down or reset. • False – boot to the OK prompt and stop
boot-command	(boot)	Command to run if auto-boot? is set to true.
boot-device	(disk)/any device alias that referees to a bootable device	Disk – the first system disk c0t0d0s0. You can choose an alternate device by name: <ul style="list-style-type: none"> • cdrom • disk2 • net
boot-file	(empty by default)	Text files to pass arguments to the operating system before it starts
diag-device	Net	Net – boot to a network device if there is a system problem.
diag-file	(empty by default)	File that has arguments if there is a system problem.
diag-switch?	False	<ul style="list-style-type: none"> • False – don't run extensive diagnostics on startup • True – run extensive diagnostics on startup.
fcode-debug	False	Used for debugging fcode.
input-device	Keyboard	Keyboard – use a keyboard if present, otherwise it defaults to serial port A.
Nvramrc	(empty by default)	Use NVRAM memory, need to set to true to use custom device aliases.
oem-banner	(empty by default)	Create a custom banner, good for copyright and security notices
oem-banner?	(false)/true	<ul style="list-style-type: none"> • False – don't display a custom banner, • frue – display a custom banner
oem-logo	(empty by default)	Display a graphical image with the oem-banner.

Variable	Values (Default)/alternate	What these values do
output-device	screen	Where to display the OK prompt, will default to serial port A if monitor is not detected.
sbus-probe-list	0123	Which sbus to probe and in what order.
screen-#columns	80	Set the width of the OK prompt screen
screen-#rows	34	Numbers of rows to display.
security-#badlogins	(empty by default)	Number of bad login attempts.
security-mode	(empty by default)/none, command, full	Security mode of the system
security-password	(empty by default)	set the security password

Table 1.2 Default NVRAM Variables

Creating Devices Using the nvalias Command

The OpenBoot PROM is designed to let a system administrator create device aliases. These device aliases can then be used in other OpenBoot PROM commands. For example, in **boot mydiskalias** the variable **mydiskalias** can be used to reference a hard drive, network adapter or any other device on the system. To create a device aliase, type the command:

```
setenv use-nvramrc? true
```

This command uses a part of the NVRAM chip to store the alias created. If this command is not typed, the device alias will be lost the next time the server loses power. It is now possible to use the nvalias and nvunalias commands to create or remove a device alias.

```
nvalias < youralias > /physical device tree
```

This command creates an alias to refer to a device.

```
nvunalias <youralias > This command would remove an alias from the system.
```

Lesson 1.4 Using the devalias command

In this lesson the readers create an alias name for the system hard drive. Because an operating system is not loaded here, the only thing happening in this lesson is the creation of the device alias variable. The alias can be safely removed from the system after the lesson is finished.

1. Power on the system.
2. Type the command **devalias**
3. Type the command **setenv use-nvramrc? true**
4. Type the command **devalias disk**
Write down the *<device_path_of_disk>* that the alias *disk* references. The path should look something like :
/pci@1f,0/ide@d/disk@0,0
5. Type the command **nvalias mydisk2 <device_path_of_disk>**
This command should look something like:
nvalias mydisk2 /pci@1f,0/ide@d/disk@0,0
6. Type the command **power-off**
7. The command **power-off** should shut down the system.
8. Type the command **devalias mydisk2**
9. Type the command **devalias disk**
Commands 8 and 9 show that the custom-made alias **mydisk2** should be the same as the **disk** alias

A device alias is a word that represents part of the symbolic tree that the SPARC hardware creates to represent all the devices on the system. The kernel picks up this device tree and uses it to load kernel modules. For example, if a workstation did not have an Ethernet card attached, why would the kernel load networking modules?

Each node on the tree is a device. The device tree looks something like:

```
/
/pci@1e,0
/pci@1a,0/pci@1/ide@3
```

The device tree is constructed depending on what devices are attached to a server. The nodes on the tree have the format

name@address:argument

name	name of the node (pci, ide, sd)
address	1f,0 2c,3 – memory location
argument	used to symbolize the first bootable slice (0)

For troubleshooting purposes, it is important to understand the basic concepts of the PROM device tree and the general format of the tree. It is not important to memorize particular node names and addresses. To see all the devices on a tree, type the command **show-devs** at the **OK** prompt.

Using the **nvedit** Command

This command is used to create device aliases, just like the **devalias** command. With some older versions of the OpenBoot PROM (version 1.x , 2.x) the **nvalias** command does not exist, and the **nvedit** command must be used.

Before the **nvedit** command can be used, type the commands:

```
setenv use-nvramrc? true
```

Now, use the **nvedit** command, like this:

```
nvedit
    devalias mydisk3 /pci@2,0/pci@3,0/ide@3/disk@3,0 (different
    depending on the hardware in the system)
    CTL + C
nvstore
reset
```

The **nvedit** command is somewhat difficult to use because it requires the following keystrokes to navigate:

- CTL C – exit the editor
- CTL U – delete the current line
- DEL – delete the current letter
- ENTER – next line
- CTL B – go back one character
- CTL F – go forward one character
- CTL P – go up one line
- CTL N – go down one line

The eeprom Command

The **eeprom** command is used inside the Solaris operating system to change OpenBoot variables. This lets the system administrator change the motherboard's settings without the need for the operating system to go down.

`/usr/sbin/eeprom` - There are different types of options that can be used with the **eeprom** command. They are:

```
# eeprom
# eeprom <PROM variable>
# eeprom <PROM variable> = New Value
```

The **eeprom** command is very handy to use on servers with 99.99% uptime requirements.

Using the Obdiag Utility

The **OBdiag** utility is an OpenBoot program designed to help the system administrator run extensive hardware diagnostics on a system. Do not break out of the operating system with the **STOP + A** command and then run the **OBdiag** utility. The **OBdiag** utility runs extensive hardware tests and could freeze the Solaris 9 operating system or the hardware. Make sure the operating system is completely shut down before running the **OBdiag** utility. Figure 1.11 shows the **OBdiag** menu.

obdiag		
1 SUNW,m64B@13	2 ebus@c	3 firewire@c,2
4 flashprom@0,0	5 floppy@0,3f0	6 ide@d
7 keyboard@2	8 network@c,1	9 parallel@0,378
10 pmu@3	11 serial@0,2e8	12 serial@0,3f8
13 usb@c,3		
Commands: test test-all except help what printenvs setenv versions exit		

Figure 1.11 The OBdiag Menu

Some of the commands shown in the last row of the **OBdiag** Menu can be used with the numbered menu items above. These commands are:

- test** To test a specific device, type the command **test #** where the number (#) is one of the numbers above. For example, to test the keyboard, an administrator would type the command **test 7**. Item number 7 is 7 keyboard@2.
- test-all** The command **test-all** is used to test all the devices listed in Figure 1.11. Be careful trying to use this command because if there is one bad device, the entire test could be frozen. If the test freezes, it's not a big problem. Just press the power button and return to the **OK** prompt.
- except** The **except** command is used to exclude certain devices from the **test-all** command. To select a device to exclude, type the command **except 4,8**. Try to exclude devices that are not being used on your system. For example, on the SunBlade 100 that is used to test Solaris 9 for this book there are no firewire ports being used for anything on the system. If the author of this book wanted to test his SunBlade 100, he would run a full **OBdiag** diagnostic on the system after typing the command **except 3**. After this command he would type the command **test-all**.

help The **help** command prints a rather short text description of the other commands. The help system is rather limited with OBDiag.

what The **what** command is used to gather extensive information about a particular piece of hardware on the system. For example, using the command **what 7** on a SunBlade 100 produces the following results:

```
Device 3 is /pci@1f,0/firewire@c,2
version : %I% %E% Copyright (c) Sun Microsystems, Inc.
name: firewire
compatible : pci108e,1102.1001
Hit any key to return to the main menu
```

printenvs This command prints values for the diagnostic configuration variables.

setenv This command sets the diagnostic configuration variables to a newly defined value. This is necessary if you need to change the location of a device in the device tree.

versions The **versions** command is used to display the version numbers of all the hardware that was found on the system. Figure 1.12 shows the output of the version command

```
1 SUNW,m64B@13          No version is specified.
2 ebus@c                1.7 00/05/30 Copyright (c) Sun Microsystems, Inc.
3 firewire@c,2         %I% %E% Copyright (c) Sun Microsystems, Inc.
4 flashprom@0,0       1.8 00/06/02 Copyright (c) Sun Microsystems, Inc.
5 floppy@0,3f0        %I% %E% Copyright (c) Sun Microsystems, Inc.
6 ide@d                %I% %E% Copyright (c) Sun Microsystems, Inc.
7 keyboard@2          No version is specified.
8 network@c,1         1.6 00/04/18 Copyright (c) Sun Microsystems, Inc.
9 parallel@0,378      %I% %E% Copyright (c) Sun Microsystems, Inc.
10 pmu@3              %I% %E% Copyright (c) Sun Microsystems, Inc.
11 serial@0,2e8       No version is specified.
12 serial@0,3f8       No version is specified.
13 usb@c,3            1.6 00/05/30 Copyright (c) Sun Microsystems, Inc.

OBDIAG LIBRARY        1.4 00/05/15 Copyright (c) Sun Microsystems, Inc.
OBDIAG MENTUOL        1.2 00/01/19 Copyright (c) Sun Microsystems, Inc.
```

Figure 1.12 Output From the versions Command

The easiest way to pull any useful information out of this command is to get the version number and the date from the second column. The date is formatted in a rather odd format YY/MM/DD. For example,

```
13 usb@c,3            1.6 00/05/30 Copyright (c) Sun Microsystems, Inc.
This has the date 00/05/30, so the date this item was created is 05/30/2000.
```

exit The **exit** command is used to exit the OBDiag utility.

Lesson 1.5 Using the obdiag Command

Type in the following commands at the **OK** prompt.

1. Power on the system.

2. Press the STOP + A keys simultaneously.
3. At the **OK** prompt, type the command **obdiag**
4. Type the command **obdiag> versions**
5. Type the command **obdiag > test 6**
The command test 6 runs a simple test on item 6, the IDE hard drive.
6. Type the command **obdiag > test 7**
The command test 7 runs a simple test on item 7, the keyboard (test 7).
7. Type the command **obdiag > exit**

Key Points to Remember

To really understand Solaris 9 it would be a good idea to purchase a SPARC-based system and a Solaris 9 Media Kit. The best used system to purchase is either an Ultra 5 workstation or a SunBlade 100. The advantage of an Ultra 5 workstation is that it is very cheap. The disadvantage of an Ultra 5 is that its memory is proprietary and somewhat expensive. The other disadvantage to an Ultra 5 is that it requires a Sun monitor or a monitor adapter to work with an SVGA monitor.

The SunBlade 100 is nice in that it has a lot of components that can be purchased directly from a manufacturer. The other benefit to a SunBlade 100 is that it has a standard SVGA adapter and can be used with almost any modern SVGA monitor. The biggest disadvantage the SunBlade 100 has is that it costs over \$1,000 brand new. If additional components are added, that price can jump up to \$ 2,000 dollars.

Think of either of these purchases more like a rental than an actual purchase. After learning Solaris 9, sell the workstation on the Internet. Most likely the workstation will bring in about the same amount of money that was spent on the workstation.

Chapter 2 Installing Solaris 9

Lessons in This Chapter

Lesson 2.1 Installing Solaris 9.....	2-37
Lesson 2.2 Creating a Generic User.....	2-37
Lesson 2.3 Graceful Shutdown of Solaris 9.....	2-39

Introduction

It is very important that the reader of this book try to set up their system as closely as possible to the recommendations described in this chapter. The partition names, disk slices, IP addresses, etc. used in this chapter will be referenced in future chapters. If the test workstation is on a network that requires different configurations, write down all the settings that deviate from this chapter's configuration.

This chapter demonstrates how to install Solaris 9 through the Web Start Installation method. The Web Start installation method is one of the easiest ways to install Solaris 9. Just put a Solaris 9 Install CDROM in the CDROM drive or the Solaris 9 DVD in the DVD drive and type the command `boot cdrom` at the **OK** prompt.

Quick Tip

- If a DVD ROM is used, different screens may be shown during the installation than the ones shown below. For example, after the initial loading screens, this message appears

hsfs mount failed, trying ufs...

mount: /devices/pci@1f,0/ide@d/sd@1,0:a is not this fstype.

- These are not real error messages; they only indicate that the hsfs (High Sierra File System) found on a CDROM is not present.
- In general, the screens that appear for a DVD installation are very similar to the ones shown below for a CDROM installation. The main difference is that you will not be asked to insert new media, such as Solaris 9 Software 2 of 2).

The Web Start installation starts with some text messages that identify the installation language, and to identify the hard drive to use to load a miniature pre-install version of Solaris 9. After the miniature version of Solaris 9 is installed on the hard drive; the system reboots itself and the Web Start installation continues in graphics mode.

The Web Start installation is very easy for a novice user to work with. Each screen presents helpful text messages that relate to the possible choices. The only difficult part is knowing what to select.

This chapter follows a typical Web Start installation, with helpful advice from the author on what are the best choices for different network setups. If the reader follows the author's setup recommendations, future lessons will be much easier to perform.

For readers that do not have Solaris 9 or SPARC-based system; it might be a good idea to read through this chapter to become familiar with the Solaris 9 installation process. More advanced methods of installing Solaris 9 will be covered in Chapter 21, Advanced Install Methods. Unfortunately there is no Intel version of Solaris 9. To work with Solaris 9, a SPARC-based system must be purchased. Read Chapter 1 for the best way to purchase a used SPARC workstation that is cheap, but powerful enough for Solaris 9.

The installation of Solaris 8 on SPARC-based systems is very similar to the installation of Solaris 9 on a SPARC system. This chapter can be used as a reference for Solaris 8 installations. The Intel version of Solaris 8 requires the use of a boot floppy know as the *Device Configuration Assistant*. Visit the Sun website <http://docs.sun.com> for documentation and source media used to install Solaris 8.

Solaris 9 Installation Methods

There are several advanced ways to install Solaris 9. Chapter 21 will describe these methods in detail. This section briefly describes the different installation methods. The advanced installation methods are:

Web Start

This is a graphically displayed installation that guides the administrator through a step-by step installation of Solaris 9. This installation uses a series of GUI (Graphical User Interface) windows that ask basic setup and networking questions. The system administrator can then choose to use a default installation or a custom installation. If a custom installation is chosen, the system administrator has to choose the software packages to load and the size of the disk's slices. If the system is not capable of producing a color video display; the Web Start installation will display all the installation questions in text mode.

Factory JumpStart Installation

This is the easiest installation method possible, but should probably be avoided. Just insert the Solaris 9 Software 1 of 2 CDROM in the CDROM drive and type the command **boot cdrom** at the **OK** prompt. Solaris 9 will ask some basic installation questions and then install the core software automatically. Unfortunately, this installation method does not allow the system administrator any choice in regard to software packages and directory sizes. This installation method has a rather bad habit of putting a lot of the disk's space into the **/export/home** directory. This causes problems because the root file system (**/**) and other critical file systems tend to be very small and useless. Avoid this installation method.

Custom Jumpstart

When a system administrator needs to install Solaris 9 on several systems, he or she can make custom Jumpstart files that will automatically answer all the questions asked during the install process. The Jumpstart files can be customized for a particular type of hardware. For example, an E-250 server could have the Apache Web server installed, and an E-450 server could have extra **man** pages installed for ordinary users. These Custom Jumpstart files can then be copied to a floppy diskette that is read during the installation of Solaris 9.

Web Start Flash

With this installation method, the administrator installs Solaris 9 on a server. After the installation, the administrator modifies the server in ways such as adding users, making directories, and changing the system's IP address. Once the final configuration of the master server is made, a Web Start Flash archive is created. This archive can then be used to make clone copies of the master server. This is very useful for creating a series of workstations that use the same configuration.

The Suninstall script

The Suninstall script is very useful for installing Solaris 9 on a system that has a minimum amount of memory. However, this installation method will not install third-party software packages.

Live Upgrade Method

This installation method lets a system administrator install an inactive version of Solaris 9 on a server that is running. This inactive copy of Solaris 9 will mimic the currently running version of the operating system. Once the installation of Solaris 9 is complete, the system administrator simply boots to the new version of Solaris 9. The previous version of Solaris is left in tact and can be rebooted if there is a major problem with Solaris 9. This drastically reduces down time associated with rebuilding a server.

Solaris 9 Media Kits

A Solaris 9 media kit contains several CDROMs, and a DVD and some documentation booklets. There are two different media kits that can be purchased from Sun Microsystems—the System Administrator’s Kit and the Slim Kit. Table 2.1 describes the CDROMs and their purpose in the installation of Solaris 9. The media kit contents might be changed in the future, so it is possible that later purchases of the Solaris 9 media kits could contain different media. This table is based on the Solaris 9 July, 2002 media kits.

Solaris CDROM	Contents
Solaris 9 Installation	Bootable CDROM, contains the Web Start install software.
Solaris 9 Software 1 of 2	Contains the Suninstall script, can be used for Factory Web Starts.
Solaris 9 Software 2 of 2	Most of the “extra software” is on this CDROM.
Solaris 9 Languages	Used for non-English installations.
Solaris 9 Documentation 1 of 2	Contains Basic Solaris 9 Documentation in English, Spanish, French, Italian and Swedish.
Solaris 9 Documentation 2 of 2	This CDROM also contains Solaris 9 Documentation.
Sun Management Center 3.0, Platform Update 4	3 CDROMs that are used to install the SMC 3.0. This is an upgrade to the default Solaris Management Console 2.1.

Table 2. 1 Solaris 9 Slim Kit Contents

Pre-installation Information

Table 2.2 shows the type of information that should be known before installing Solaris 9 on a workstation. All the required installation information will be presented later in the chapter. Table 2.2 provides as a general reference for the types of information required for an installation.

Question	Possible Answers	Author’s Suggestion
Will the system be networked	Yes/No	This should be yes in almost all cases.
Use DHCP	Yes/No	Requires a DHCP server on the same network segment.
Hostname	2-8 characters	Required, use the command hostname on the old system before installing Solaris 9.

Question	Possible Answers	Author's Suggestion
IP Address	xxx.xxx.xxx.xxx x = 0-9 digit	Obtain from ISP or Network Engineer. Type cat /etc/hosts to see the current settings.
Subnet Mask	255.255.255.xxx x = 0 - 9 digit	Most common type is C address 255.255.255.0 Type the command cat /etc/netmasks to see the current settings.
IPv6	Yes/No	Obtain IPv6 info from the Network Engineer.
Kerberos	Yes/No	Requires Kerberos server on Network, rare.
Name Service	NIS+,NIS,DNS,LDAP	On small networks this is not needed, on large networks, one of these will exist. Type the command cat /etc/nsswitch.conf to see the current settings, and record the information.
Default Router	Router's IP address	Obtain the router's IP address before installing Solaris 9.
Time Zone	Geographic region, oOffset from GMT, Time Zone file	Use Geographic region if only in one location; use GMT for international servers.
Proxy Server	Yes/No, Proxy server's hostname, IP and port #	If a proxy server exists, check Netscape navigator for the proxy server's setting on the old system before installing Solaris 9.
Software Group	End User, Developer, Entire Solaris Software Group, Entire Solaris Software Group plus OEM	Different types of software installed' use Entire Solaris Software Group if possible.

Table 2. 2 Pre-installation Information

Solaris 9 has the following minimum system requirements:

Minimum Required

SPARC
96 MB RAM
4 GB hard drive
Sun Keyboard + Mouse
Generic SVGA monitor

Author's Recommendation

SPARC
256 MB RAM
Dual hard drives, minimum 5 GB each
Sun Keyboard + Mouse
17" SVGA monitor

The author's system is

SunBlade 100
384 MB RAM
128 MB SDRAM with ECC
256 MB SDRAM with ECC

Dual Hard Drives

System disk: 4.0 GB Western Digital (installed a small hard drive to make sure the lessons can work with a tiny hard drive.)

Second disk: original 14 GB Seagate.

DVD drive

Sun Keyboard + Mouse

17" SVGA Samsung Monitor

U.S. Robotics 56K external modem

Displaying the OK Prompt

This section describes how to display the **OK** prompt on a SPARC system. There are four possible situations:

Situation 1) The workstation is powered off. In this case, push the power button. If the OpenBoot variable **auto-boot?** is set to true, the workstation will start to boot the operating system. Press the **STOP + A** keys simultaneously (several times, until the **OK** prompt appears) to halt the boot process and display the **OK** prompt. Understand that the POST (Power On Self Test) does not recognize the keyboard in the first one or two seconds of its routine. For systems with a PC-style keyboard attached, press the keys **CTL + BREAK** simultaneously.

Situation 2) The workstation is powered off. In this case, push the power button. If the OpenBoot variable **auto-boot?** is set to false, the workstation will perform an initial power on self test and halt at the **OK** prompt. The operating system will not automatically start.

Situation 3) The operating system is running, with the CDE display. In this case, right click anywhere in unoccupied desktop space. The Workspace menu, as illustrated in Figure 2.1, will appear at the click point.

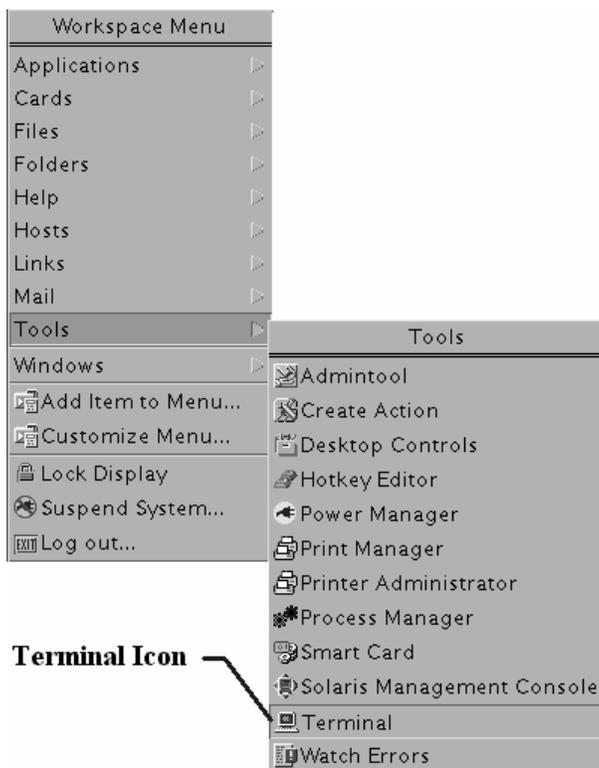


Figure 2. 1 The Workspace Menu

Left click on the Tools menu choice, then left click on the Terminal Icon. When a Terminal window appears, type the command **init 0** (as the root user). This command tells the Solaris 9 operating system to gracefully shut itself down. Once the operating system is gracefully shut down, the **OK** prompt will be displayed.

WARNING

If the workstation is running, the STOP + A keys should not be pressed to bring the system to the **OK** prompt. The problem with using STOP + A on a running system is that the operating system will not shut down properly, and the file systems will become badly damaged.

Situation 4) The workstation is powered on, and the operating system is running with a text only display. In this case, type the command **init 0** (as the root user).

Quick Tip

- If no video is displayed when the workstation is powered on, check the monitor's power plug and the video cable. The SunBlade 100 connects directly to a standard SVGA monitor, so check the video cable connection.
- Some older monitors don't display the **OK** prompt in a graceful manner. They might skew the text or wrap the text in an odd manner. It is a real hit and miss thing when it comes to older monitors if the **OK** prompt's characters look nice. Regardless of the beauty of the display, the information displayed is the same.
- The cheapest monitors to use with the SunBlade 100 and Ultra 5 workstations are the Samtron™ monitors. They can be picked up from Best Buy™ for a very cheap price and they work really well with Sun equipment.

Just as a quick side comment. The author of this book does not receive product placement revenue from mentioning Best Buy, Samtron or any other product mentioned in the book. This book is written for your benefit, with honest advice.

The **OK** prompt is used to display and work with OpenBoot variables. If the OpenBoot variable **auto-boot?** is set to **false**, the system will perform the initial hardware diagnostics and then stop at the **OK** prompt. If the **auto-boot?** variable is set to **true** the system will perform the initial hardware diagnostics and then try to boot the operating system on the default boot device.

To change the **auto-boot?** variable to false, type the command:

```
setenv auto-boot? false
```

Type the command **printenv**. This command shows all the OpenBoot variables. Chapter 1 describes in detail how to work with OpenBoot commands.

Once the system is at the **OK** prompt, insert the Solaris 9 Install CDROM into the CDROM drive. If the Solaris 9 Operating Environment DVD is being used, insert the DVD into the DVD drive. After the install media is inserted, type the command **boot cdrom** from the **OK** command prompt.

Lesson 2. 1 Installing Solaris 9

This lesson shows how to perform a Web Start installation. For this lesson, the reader needs to obtain the Solaris 9 CDROMs and a SPARC-based computer with a CDROM drive.

Quick Tip

The following text format is used to describe the installation of Solaris 9 :

commentary in italics

screen text in ordinary text

commands you should type in bold wide text. Some commands and variables are also in bold wide text..

WARNING

WHEN YOU INSTALL SOLARIS 9 FOLLOWING THE DIRECTIONS IN THIS LESSON, THE HARD DISK WILL BE REFORMATTED AND ALL EXISTING DATA WILL BE ERASED. IF YOU HAVE CRITICAL DATA THAT MUST BE PRESERVED, BACK UP THAT DATA TO A SECURE MEDIUM BEFORE YOU CONTINUE WITH THIS LESSON.

*To start the installation, gracefully bring the system down to the **OK** prompt. Once the system is at the **OK** prompt, insert the Solaris 9 Install CDROM in the CDROM drive. If a DVD drive is available, the only disk needed during the installation is the DVD drive itself.*

If a monitor is not directly connected to the workstation, the installation can be performed with a null modem cable connected to serial port A. Chapter 1 describes how to use a null modem cable with a SPARC-based system. The text installation will be performed with a somewhat difficult-to-use text version of the Web Start installation instead of the nicer GUI window screens. Fortunately, these text screens present the same information as the GUI windows.

Type the command **OK boot cdrom**

*Before the installation of Solaris 9, a miniature copy of Solaris 9 is put into memory. This version hosts the **Suninstall** program. The next couple of lines are related to the loading of the miniature version of Solaris 9. During the loading of the miniature Solaris 9, the (/) symbol is spun around. This lets the administrator know that the installation is continuing. If the (/) symbol stops spinning, the installation has hung for some reason and the system administrator should try the installation again. Please note, Sun CDROMs have a rather bad habit of*

stalling during the initial install. The author of this book recommends that if the installation stalls before the GUI part of the install begins, just power down the system and try again.

Quick Tip

- SPARC systems sometimes have problems working with CDROM drives and CDROM disks during the installation. There will be times when you will have to open and close the CDROM drive several times to get Solaris 9 to properly read the CDROM. Each time the CDROM drive is opened and closed it sends a signal to the install program to re-read the CDROM. Once Solaris 9 is installed, it works just fine with CDROMs and DVDs. It's just getting through the installation that is a problem.
- Make sure the CDROM is clean and free from scratches. If the CDROM is damaged it will cause a lot of problems for the Solaris 9 installation program.
- The nice thing about a DVD drive is that you don't need to swap CDROM disks several times. Purchase a LiteOn DVD drive and replace the stock CDROM drive with a DVD.

The following text is displayed on the screen before the installation starts

Initial moving (/) symbol

The slash (/) symbol is rotated during the installation to indicate the system has not stalled. If the slash (/) character stops spinning for a significant amount of time, that means the system is frozen. Reboot the system and try the installation again. If the system continues to freeze during the installation, there might be a hardware problem.

SunOS Release 5.9 Version Generic 64-bit
Copyright 1983-2001 Sun Microsystems, Inc. All rights reserved.
use is subject to license terms.
Configuring /dev and /devices

*The **/dev** and **/devices** directories are configured with system information. These two directories hold system information. For example, a modem will be represented by a file in the **/dev/cua** directory, and a tape drive will be represented by a file in the **/dev/rmt** directory. The text "Configuring /dev and /devices" is shown when Solaris 9 creates configuration information about the system.*

*If the installation freezes when trying to create the /dev and /devices directory, check the systems hardware. The **OK** prompt has an excellent diagnostics tool named **OBDDiag**. To use **OBDDiag**, just type the command **obddiag** and follow the menu item. This is a rather straightforward diagnostics tool.*

moving slash (/) symbol will be seen from time to time.

Using RPC Bootparams from network configuration information.

The line about RPC (Remote Procedure Call) Bootparams deals with network installations of Solaris 9. Ignore this line for now. This is not a network installation.

Skipping interface eri0

This installation was performed on the author's SunBlade 100. The SunBlade 100 has an Ethernet card with the device name eri. Other systems have Ethernet card names like hme or qfe. The zero (0) at the end of eri0 is used to signify that this is the first Ethernet card. If a second Ethernet card was installed, it would have a name like eri1. The "Skipping interface eri0" line just means that the Solaris 9 installation is ignoring the Ethernet card for now.

Searching for configuration files (s)...

Searching for configuration file(s)...

The Solaris 9 Operating Environment is looking for Jumpstart installation files. If it can not find Custom Jumpstart installation files; it just uses the default files on the CDROM.

Search complete

The installation has two major steps. The first step is text based. The second part of the installation is GUI based and starts when the system reboots for the first time. The text part of the installation is now starting. The CDROM and DVD disks produce a slightly different set of messages during the installation; just use common sense to follow this lesson. This lesson is based on a CDROM installation.

If a CDROM is being used for the installation, the following message will appear:

Solaris Installer can be run in English, or any of the following languages:

- | | |
|------------|-------------------------|
| 1) English | 6) Japanese |
| 2) German | 7) Korean |
| 3) Spanish | 8) Swedish |
| 4) French | 9) Simplified Chinese |
| 5) Italian | 10) Traditional Chinese |

Select the language you want to use to run the installer:

Select Item 1) English

If a DVD is being used, the following message will appear:

Select a Language

- 0. English
- 1. French
- 2. German
- 3. Italian
- 4. Japanese
- 5. Korean
- 6. Simplified Chinese
- 7. Spanish
- 8. Swedish
- 9. Traditional Chinese

Please make a choice (0-9), or press h or ? for help:

Select 0) English

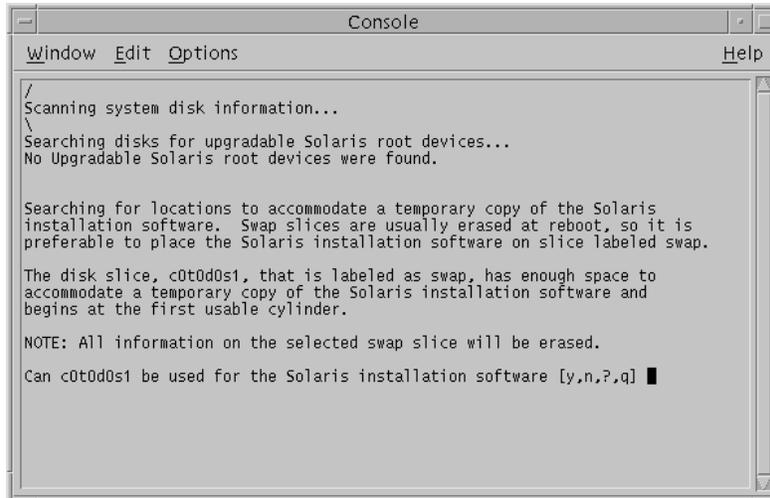
Select a Locale

- 1. Russia
- 2. Chinese
- etc...

Select 58. U.S.A. (en_US.ISO8859-15)
This is the best choice for a language Locale.

Starting the
Common Desktop Environment
CDE Version 1.50

After the installation language is chosen, the CDE (Common Desktop Environment) is started. CDE 1.50 is used for the setup of Solaris 9. Now a Console window opens with the following message



```
Console
Window Edit Options Help
/
\ Scanning system disk information...
\ Searching disks for upgradable Solaris root devices...
No Upgradable Solaris root devices were found.

Searching for locations to accommodate a temporary copy of the Solaris
installation software. Swap slices are usually erased at reboot, so it is
preferable to place the Solaris installation software on slice labeled swap.

The disk slice, c0t0d0s1, that is labeled as swap, has enough space to
accommodate a temporary copy of the Solaris installation software and
begins at the first usable cylinder.

NOTE: All information on the selected swap slice will be erased.

Can c0t0d0s1 be used for the Solaris installation software [y,n,p,q] █
```

If a previous version of Solaris is installed on the system, this message would appear.

Release	Root Device
Solaris 8	c0t0d0s0

Most installations will not show the Solaris8 upgrade text. If a server or workstation has a previous version of Solaris installed, Solaris 9 tries to install itself in the swap partition. This is a safety precaution, so that if the installation of Solaris 9 goes bad, the original operating system can be rebooted without any damage. The swap partition is erased when Solaris starts, so the bad installation is automatically cleared out and the previous operating system can be run again.

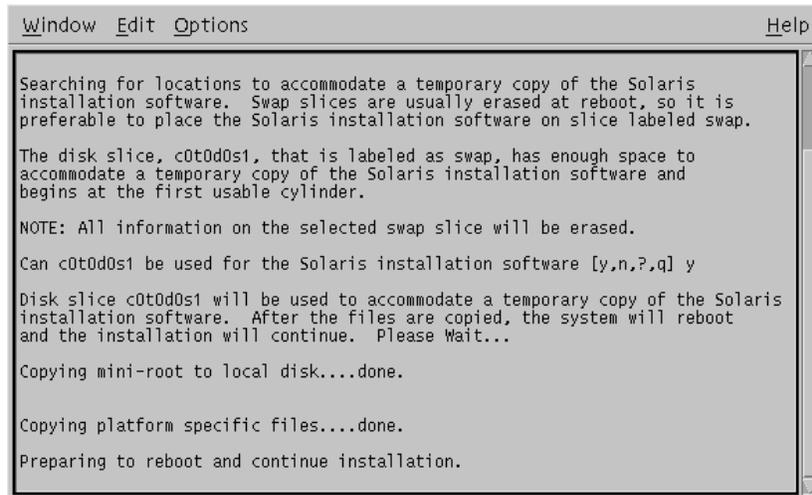
A smaller Console Window is located at the bottom of the screen, with the root prompt# symbol. If extra commands need to be run during the installation, they can be run from this window.

This comment is for senior level Solaris Administrators

What happens next is rather interesting. A miniature copy of the Solaris 9 operating system is installed on the c0t0d0s1 slice. This slice is usually the swap slice. The nvramrc OpenBoot variable is set to devalias mydisk2 /pci@1f,0/ide@d/disk@0,0 true.

Also, the boot-device is set to disk: a . The combination of these variables boots off the system disk or the first slice when a SPARC system reboots. It does not reboot from the CDROM on the reboot.

The final screen should look like this:



You will also see the lines:

rebooting the system

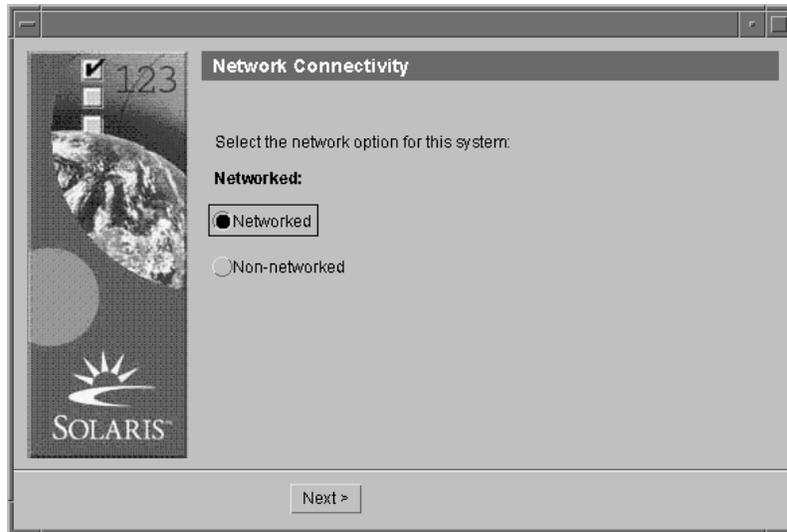
Rebooting with command: `boot /pci@1f,0/ide@d/disk@0,0:b (slice 1 is :b)`

After some software is loaded onto the hard drive, the system is rebooted.

The Web Start Installation starts in graphical mode, A CDE welcome splash screen now appears with the message "Welcome" and the screen is replaced with a blue background.

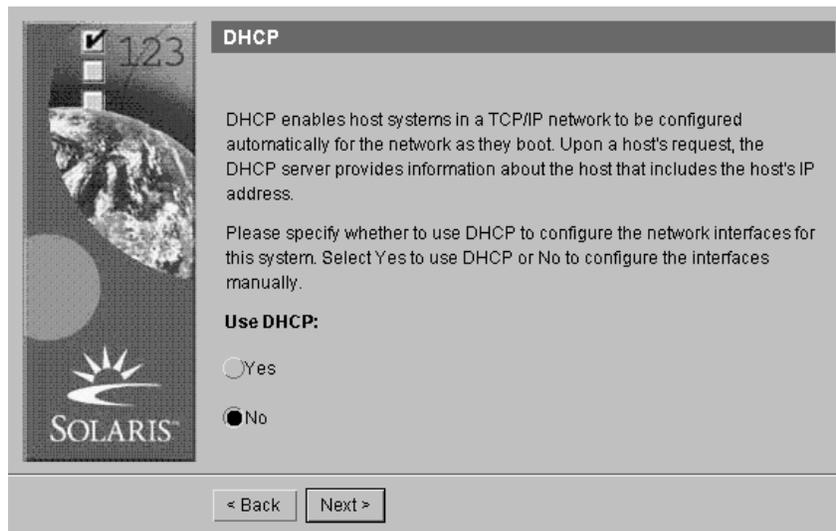


Left click on the Next > button to advance to the next screen.



If a server has an Ethernet card attached, select Networked. All new Sun servers have an Ethernet card built into the motherboard. If for some reason network connectivity is not wanted, select Non-networked. Understand it will be very difficult to reconfigure an operating system to have network connectivity if this option is not selected now.

Left click on the Next > button to advance to the next screen.



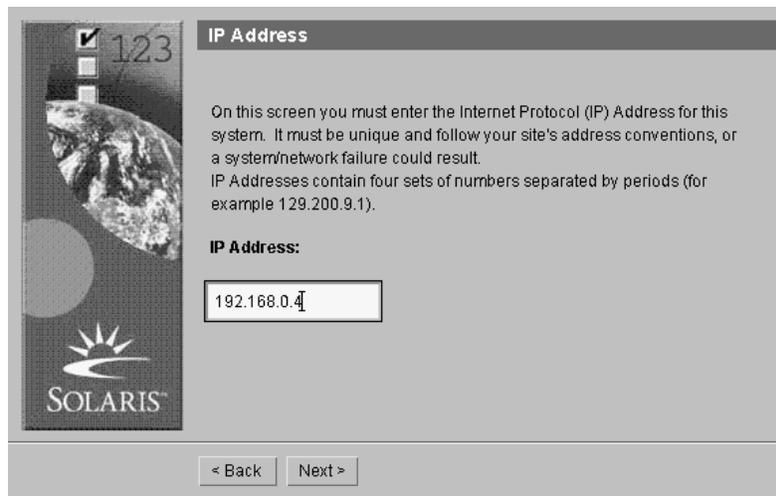
DHCP (Domain Handling Control Protocol) requires a DHCP server be present somewhere on the same network as this system. If you don't have a DHCP server up and running on your network, do not select this option. If this computer is being used as a production level server (instead of a desktop system) do not use DHCP. Workstations use DHCP; servers never use DHCP.

*Select **No**. Then left click on the Next > button.*

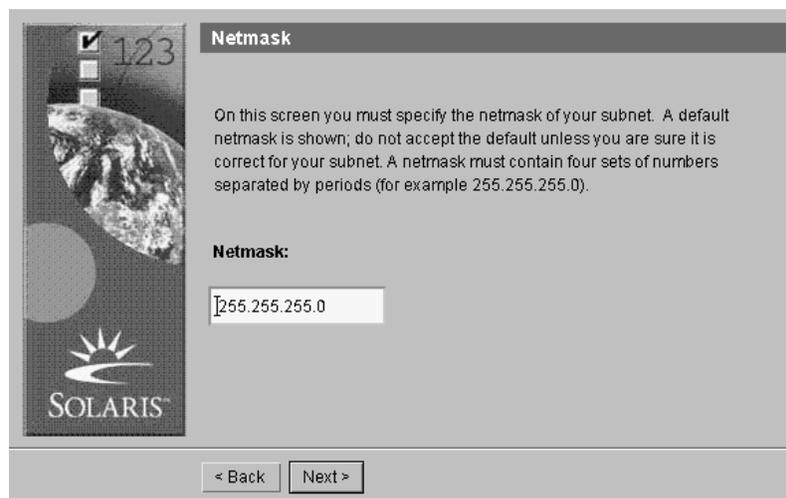
Notice that there is a <Back button at the bottom of the screen. From here on, if you make a mistake entering or selecting information in any screen, just click the <Back button. This will return you to that screen. You can then correct the information before moving on.



Enter **sun100** as the Host Name. Then left click on the Next > button.

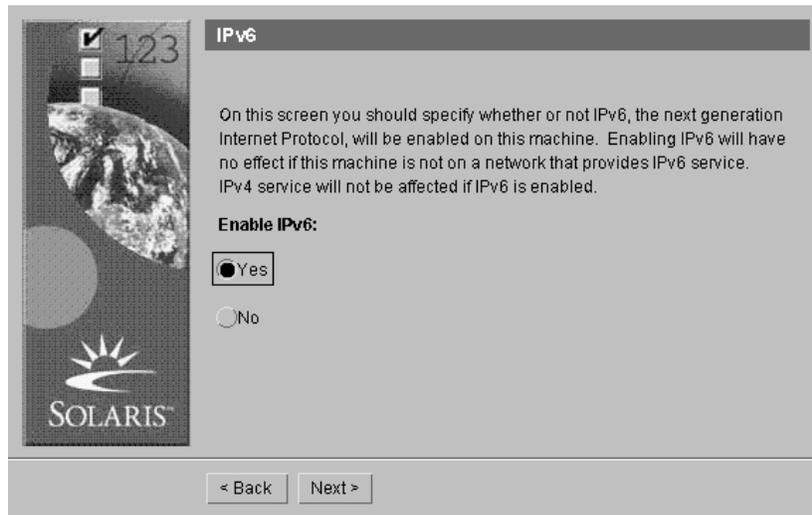


Enter **192.168.0.4** as the IP Address. Then left click on the Next > button. .



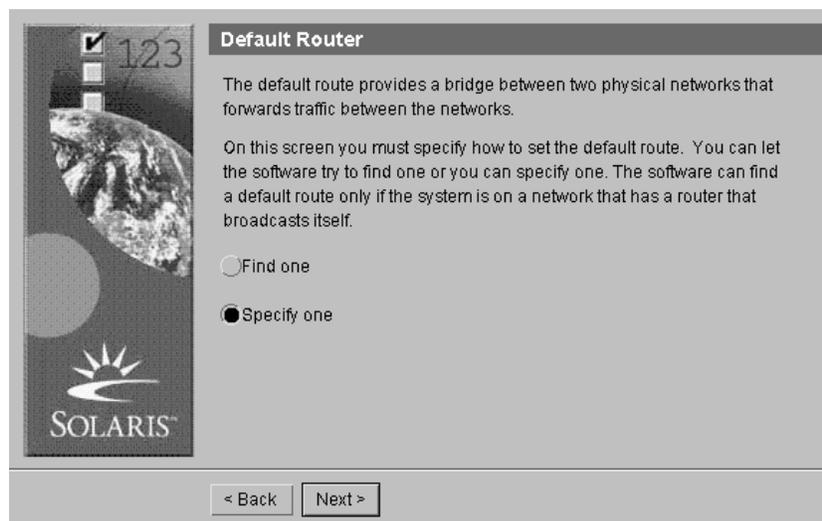
An IP address has two components, a network address and a host address. This is similar to a street address such as 1235 Johnson Avenue. The number 1235 represents the house and Johnson Avenue represents the road. An IP address 192.168.0.4 represents the network 192.168.0 (Johnson Avenue) and the 4 represents the host address 4 (house 1235). Because two pieces of information are in the IP address a netmask shows where the network address (Johnson Avenue) starts and where the host ID (1235) starts.

Enter the Netmask as 255.255.255.0. Then left click on the Next > button..



IP version 6 was supposed to replace IP version 4, starting in the year 2000. Unfortunately, most companies and networks have not transitioned to IP version 6 yet. Contact your network engineers if IP version 6 is needed.

*Select **Yes**. Then left click on the Next > button.*



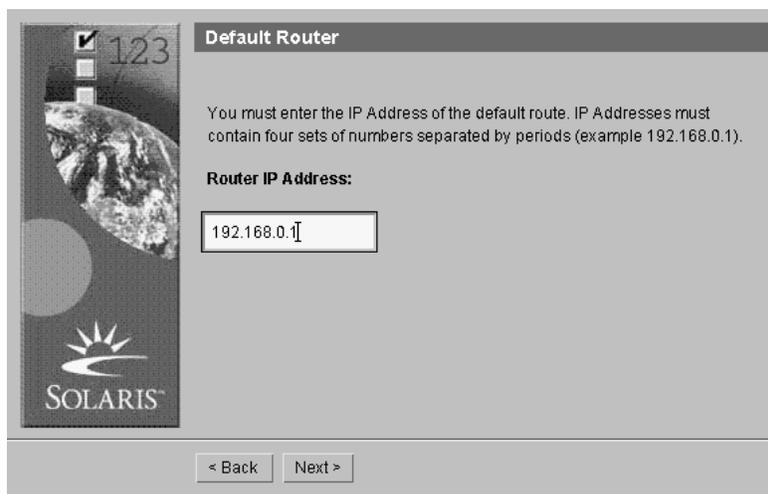
On this screen you must specify how to set the default route. You can let the software try to find one, or you can specify one. The software can find a default route only if the system is on a network that has a router that broadcasts itself. Most companies hard code the router's information into a server. If the install program can not find a router, it will stop the installation of Solaris 9.

It's best to type in the router's IP address and not use the router discover protocols. Please note that default router = default route = default gateway = gateway = router = router IP address. All these terms mean the same thing!

A router is a piece of network equipment that is used to connect separate network segments together. This screen is asking you for your router's IP address. If you have a home network that does not include a router choose "() find one " as an option. Most companies demand that a router is specified with an IP address. You can still contact other computers on the same network segment as this Solaris server if they have the same netmask and are in the same IP range. You just can't connect to the Internet or other computers outside your network without a router.

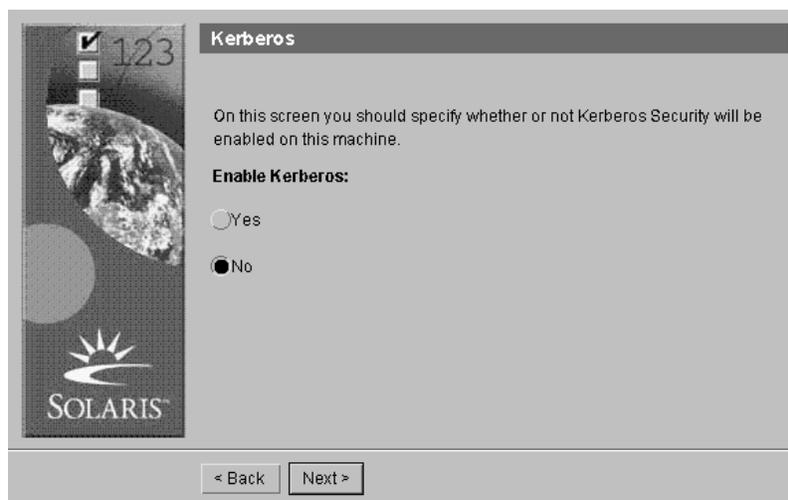
*Select **Specify One**. Then left click on the Next > button.*

If you selected Specify One, this screen will be shown:



Next >

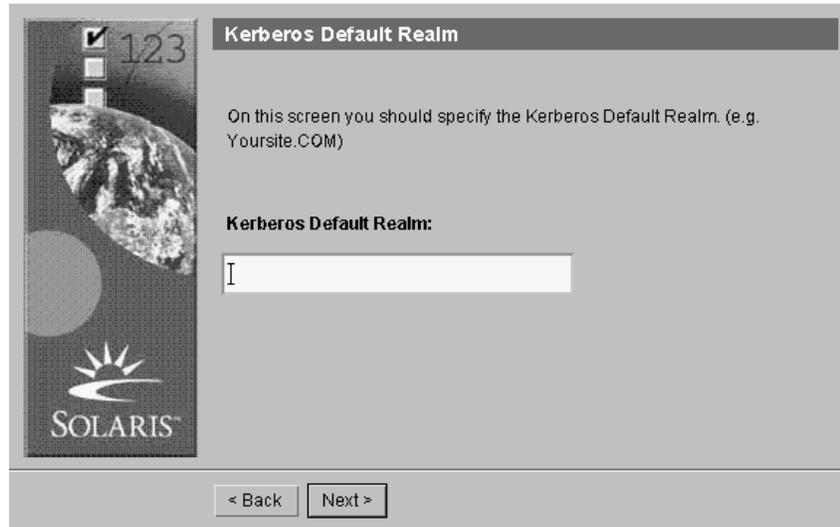
*Enter **192.168.01** as the Router Address. Then left click on the Next > button.*



This screen is used to specify whether or not Kerberos Security will be enabled on this machine. A Kerberos security server must be present to use this option.

Select **No** for Kerberos. Then left click on the Next > button.

The Kerberos Screen looks like this (if No was selected, the screen does not appear. This is being shown just for reference` To see this screen, select Yes above.)



The screenshot shows a window titled "Kerberos Default Realm". On the left is a vertical sidebar with a checkmark and the number "123", a globe image, and the "SOLARIS" logo. The main area contains the text: "On this screen you should specify the Kerberos Default Realm. (e.g. Yoursite.COM)". Below this is a label "Kerberos Default Realm:" followed by an empty text input field. At the bottom are two buttons: "< Back" and "Next >".

Next >

If you see this screen, enter the Kerberos Default Realm in the field shown Then left click on the Next > button.



The screenshot shows a window titled "Name Service". On the left is a vertical sidebar with a checkmark and the number "123", a globe image, and the "SOLARIS" logo. The main area contains the text: "On this screen you must provide name service information. Select the name service that will be used by this system, or None if your system will either, not use a name service at all, or if it will use a name service not listed here". Below this is a label "Name Service:" followed by four radio button options: "NIS+", "None", "NIS", "DNS", and "LDAP". The "None" option is selected. At the bottom are two buttons: "< Back" and "Next >".

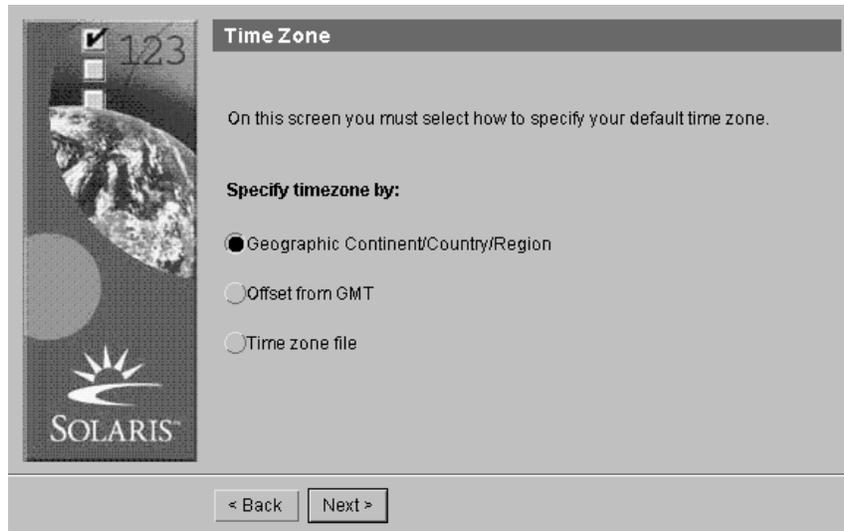
Name Service

On this screen, you must provide name service information. Select the name service that will be used by this system, or None if your system will either not use a name service at all, or if it will use a name service not listed here

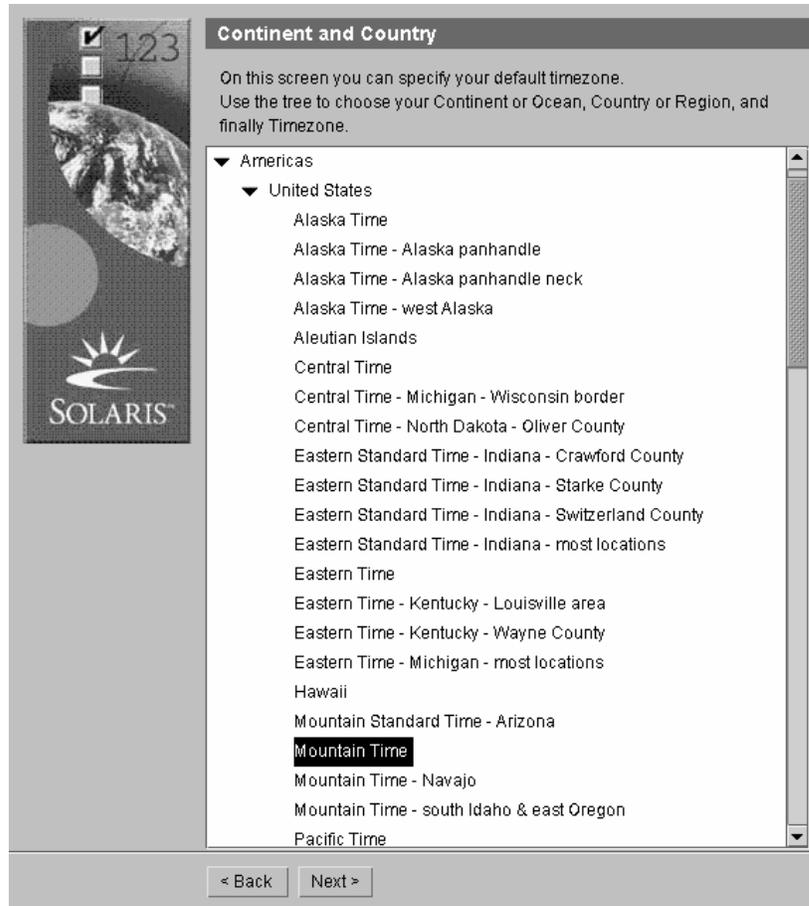
Select the appropriate Name Service, or **None**. Then left click on the Next > button. If a mistake was made, click on the < Back button to return to the previous screen.

One of the functions of a name service is to match a computer's name with an IP address. For example, if a user types the command **ping dataserver.Sun.com** the user is trying to send a test IP packet to another server named "dataserver" on the Sun.com network. This computer needs to match the hostname "dataserver" with an IP address. It first tries to look up the hostname "dataserver" with an IP address from its internal files. If it can not find this IP address, it tries to find it from a NIS+, NIS, DNS or LDAP server on the network. Unless at least one of these servers is present on your network, select theNone option.

Select the appropriate Name Service, or None. Then left click on the Next > button.



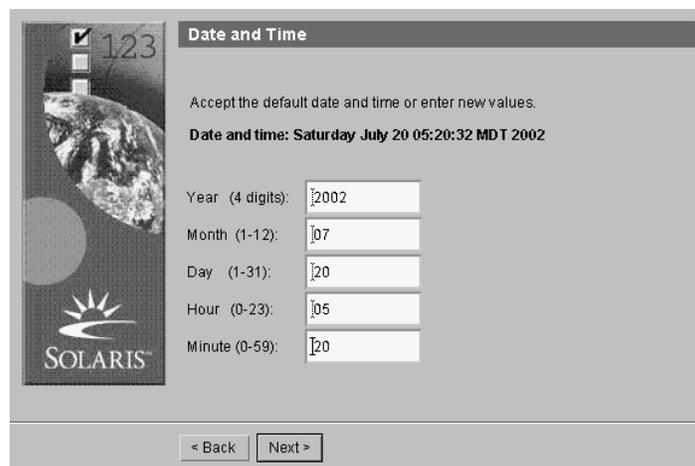
Select **Geographic Continent/Country/Region**. Then left click on the Next > button.



If the menu choice **Offset from GMT** is selected, a screen appears that lets the user scroll a button on a slider left or right to add or subtract hours from Greenwich Mean Time.

If the menu choice **Time zone file** is selected, a screen appears that lets the user specify a default time zone file in the `/usr/share/lib/zoneinfo` directory. The administrator does not need to specify the path, only the file name is necessary.

Select the time zone (in this example, **Mountain Time** was chosen). Then left click on the **Next >** button.



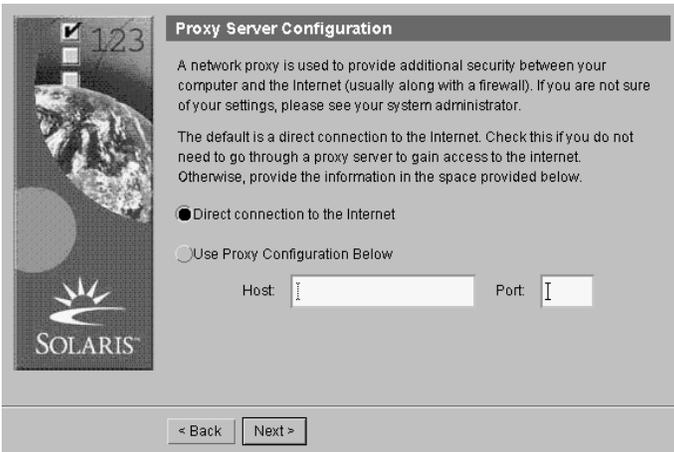
The install program reads the motherboard's date and time and offers those as the selection. Each field has a text box where you can enter a different value.

Change the date and time information as needed. Then left click on the Next > button.

The screenshot shows the 'Root Password' configuration window in the Solaris installer. On the left is a vertical sidebar with a globe, the number '123', and the 'SOLARIS' logo. The main area has a title bar 'Root Password' and instructions: 'Type in an alphanumeric string to be used as the root password for the computer you are setting up.' Below this is a text box containing '*****'. The next instruction is 'Retype the above password for confirmation.' followed by another text box containing '*****'. At the bottom are '< Back' and 'Next >' buttons.

This screen is used to create the root user's password. The root user has the name "root" and is the most powerful user on the system. Write down this password. The root password should be at least 8 characters long and should include some numbers.

Enter the password in the text boxes. Then left click on the Next > button.

The screenshot shows the 'Proxy Server Configuration' window. The sidebar is identical to the previous screen. The main area has a title bar 'Proxy Server Configuration' and explanatory text: 'A network proxy is used to provide additional security between your computer and the Internet (usually along with a firewall). If you are not sure of your settings, please see your system administrator. The default is a direct connection to the Internet. Check this if you do not need to go through a proxy server to gain access to the internet. Otherwise, provide the information in the space provided below.' There are two radio buttons: 'Direct connection to the Internet' (which is selected) and 'Use Proxy Configuration Below'. Below the second radio button are 'Host:' and 'Port:' labels with corresponding text input boxes. At the bottom are '< Back' and 'Next >' buttons.

The default Proxy Server Configuration is a direct connection the Internet. Check this if there is no proxy server on the network. If a proxy server is used to access the Internet, provide the hostname and port or the IP address and port of the proxy server n the box. A Proxy server is used to cache Internet files and provide extra security between your computer and the Internet. Don't select a proxy server unless your network has a proxy server present.

Click **Direct Connection to the Internet**, or enter the proxy server information. Then left click on the Next > button.



This screen confirms all the previous choices. Nothing can be changed on this screen—it's just a summary screen.

If any of the information is incorrect:

1. *Click the <Back button until you come to the screen where the mistake was made.*
2. *Change the information as needed, and then left click the Next > button until you return to this screen.*

*Select **Confirm** when you are finished. You will see the message*

Please wait while the system's information is updated...

After some files are copied, a new phase of the Web Start Installation begins. The Ethernet card is now initialized, and Internet connections can be run from the various application windows.

You will now see the screen shown below. Here's what's on it:

- *The installation process will continue in the center area of the screen.*
- *On the left side of the screen is a menu that lets you call up optional documentation and utilities, as described in the Quick Tip below. You do not need to use these; if you wish, you can continue the installation from this point.*



Quick Tip

- From this point on, while the installation is doing long operations such as file copies, you can read additional information on Solaris 9, by using the menu on the left side of the screen. (You do not have to use any of these hyperlinks or utilities.) See below for information on these menu options.
- You can also run some additional utilities, such as a Terminal or Console window.
- To view the documentation screens or use these utilities, left click on the small orange triangle to the left of an item. A drop-down menu will appear below the menu items. Left click on any hyperlink to read the article or use the utility.

The menu items on the left side are listed below.

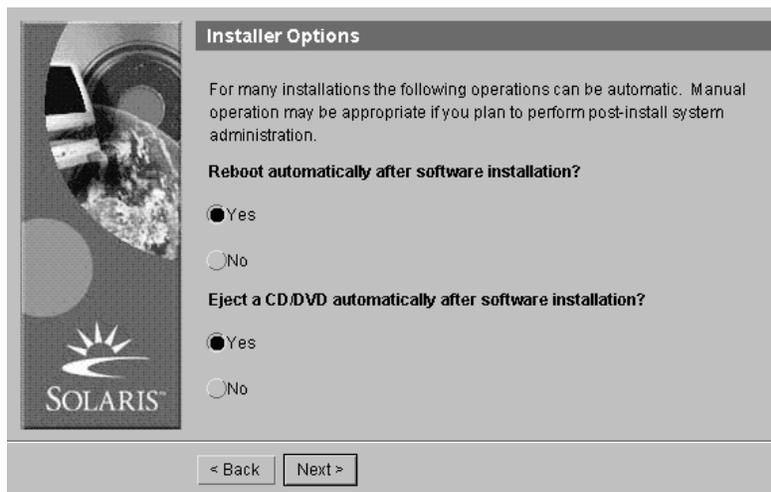
What's New	<i>This provides summary information on what new features were provided with previous versions of Solaris 8 and Solaris 9. It presents small chapters on various topics like Networking, Development Tools, and Java Releases. etc.</i>
Solaris Technologies	<i>In-depth documentation on Solaris technologies and applications like Solaris Web Start software, Solaris JumpStart technology, and Sun Management Center 3.0 Software.</i>
Additional Software	<i>Documentation on additional Software CDs that come with some media kits. The additional software includes the Star Office CD and the Solaris Software Companion CD.</i>
Bookmarks	<i>Various Internet WebPages like www.Sun.com, www.yahoo.com, and www.netscape.com. For some reason, these hyperlinks and a browser can be pulled up from here. Don't ask why www.yahoo.com is included in this list.</i>
Documentation	<i>Release Notes - The latest information about installing Solaris 9 Sun Documentation - Various documents from Sun Microsystems. If an Internet connection can not be established, a window pops up, indicating the error message.x</i>
Applications	<i>Some tools that can be run during the installation. These are: Terminal - Opens a terminal window as the root user Console - Opens a console window as the root user Clock - Shows an analog clock Browser Window - Opens up a copy of the Netscape browser</i>
Kiosk Utilities	<i>Send Kiosk to background</i>

A kiosk is a terminal or monitor that is set up for public display and does not have a keyboard. For example, many airports use a kiosk display to show flight information. The Kiosk utilities will not be covered in this book.

As mentioned above, you can continue the installation process at the center of the screen, whether or not you use the menu at the left. A small window will appear in the center of the screen with the following message:



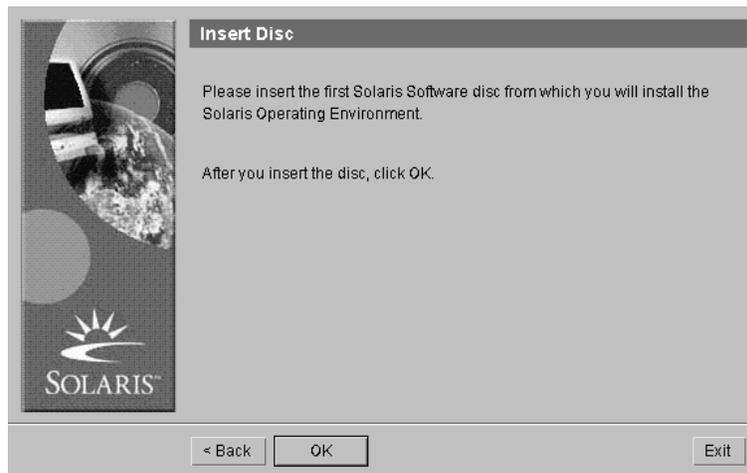
Left click on the Next > button.



*Select **Yes** for both of the options. Then left click on the Next > button.*

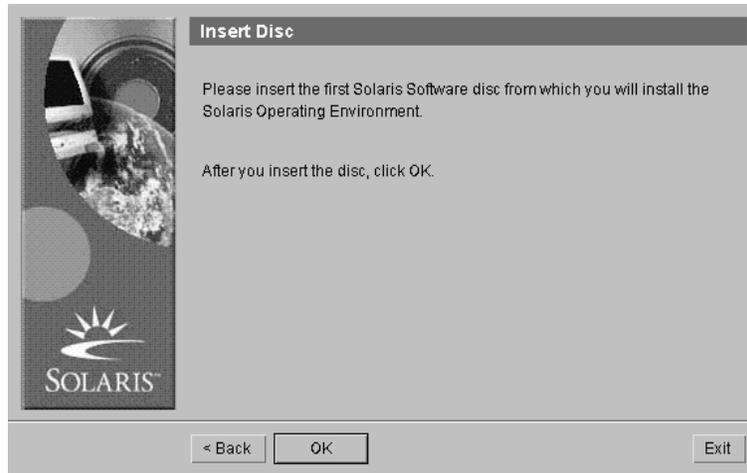


Select **CD/DVD**. Then left click on the *Next >* button.

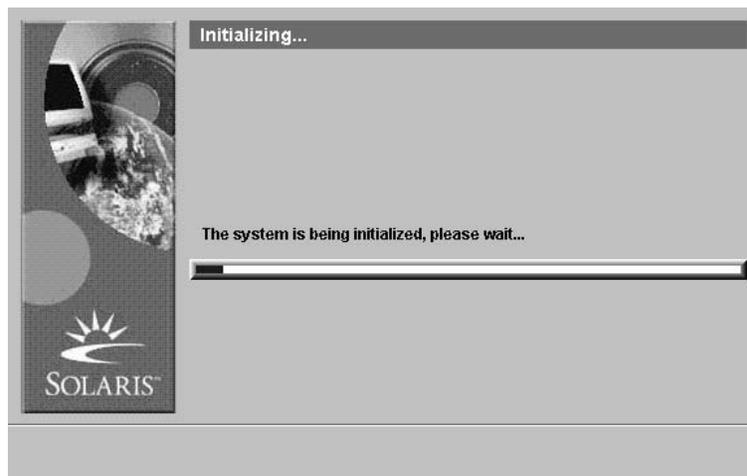


Insert the Solaris 9 Software 1 of 2 CDROM. Then left click the *OK* button.

Sometimes the installation program will not recognize a CDROM or DVD and will not let the user open the CDROM drive to insert a new one. Unfortunately, the only real option then is to restart the installation. You will just need to be patient with the problem.



Insert the first Solaris Software disc for installing the Solaris Software Environment. Then left click on the OK button.



This screen is for information purposes only. There are no buttons to choose from on this screen.

WARNING

If the system had a previous version of Solaris 8, this message will appear:

Select Upgrade or Initial install

To perform an initial install of the Solaris Operating Environment select Initial Install. Existing contents of your hard disk will be destroyed.

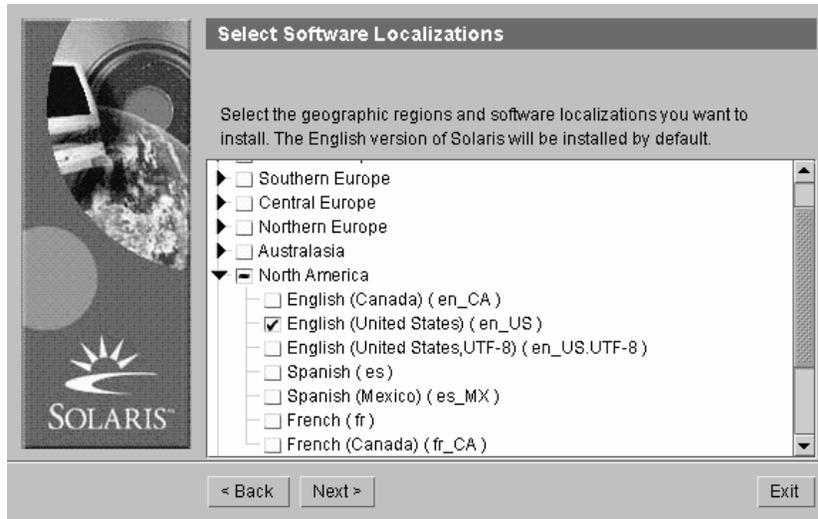
To perform an upgrade of the Solaris Operating Environment select Upgrade. The Solaris Operating Environment will be updated and other contents on the hard disk will be left unchanged. Backing up the existing Solaris Operating Environment is recommended.

Upgrading Solaris is a very bad idea! Most of the time the upgrades will stall out during the upgrade process, leaving you will a dead server. Also, an upgrade tends to leave plenty of old files on the hard drive, along with some problems. Also, most commercially available software is compiled for a specific version of Solaris, so some of your software could stop working with the new version of Solaris.

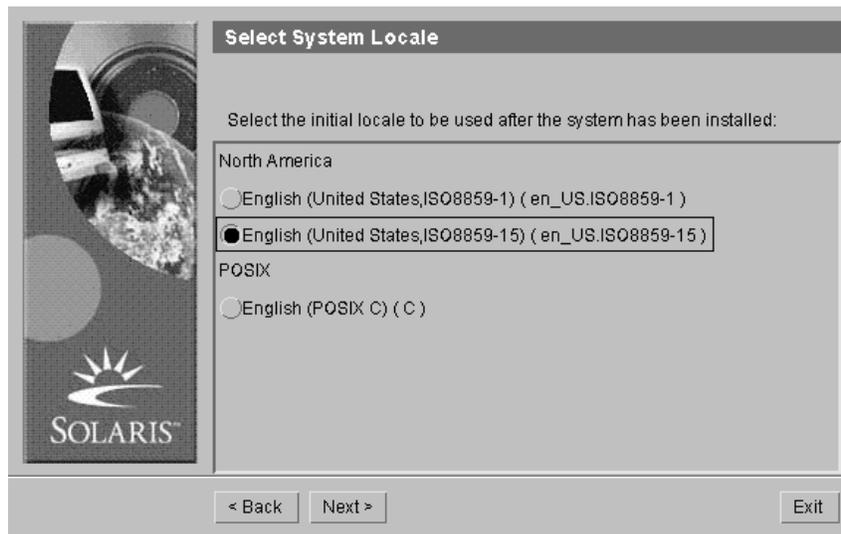
The best advice is to clean off your hard drive and start with a fresh copy of the Solaris 9 Operating System. Then, re-install software compiled for Solaris 9. Solaris 9 will upgrade the three previous versions of Solaris: Solaris 2.6, Solaris 7 and Solaris 8. Don't ask why Sun Microsystems went from Solaris 2.6 to Solaris 7. Everyone is scratching their heads about that.



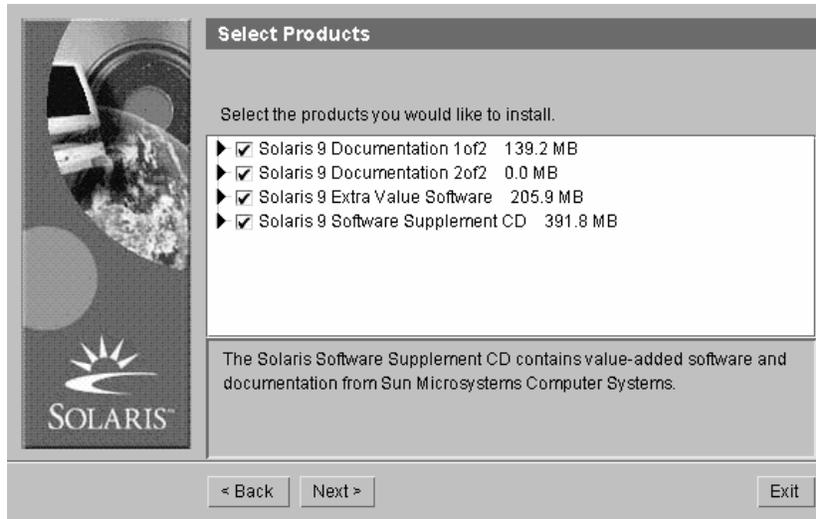
*Select **Custom Install** and then left click the Next > button. The default installation is bad because it tries to put as as much space as possible in the /export/home directory. It is a much better idea to use the Custom Install option and manually partition the disks yourself.*



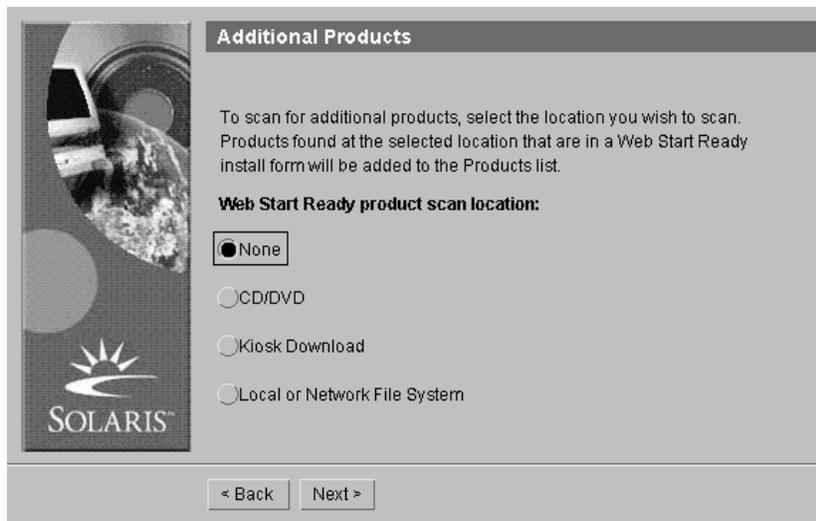
Select **English (United States) (en_US)** and then left click the **Next >** button.



Select **English (United States,ISO8859-15) (en_US.ISO8859-15)** and then left click the **Next >** button.

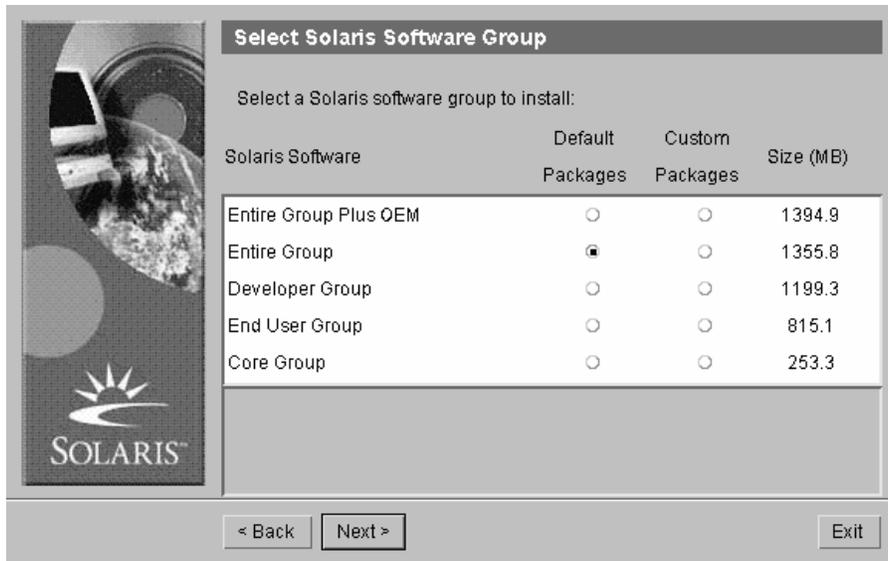


Select the products that you want to install and then left click Next >. Select all of these only if the workstation's hard drive is big enough. A 5 GB hard drive should be able to handle this extra information. The Solaris 9 Software Supplement CD adds a lot of software onto the hard drive that works with E-10K and other high end servers. This tends to slow down the operating system on a workstation without adding anything of real value to the workstation.



If you have additional software for your system, this screen lets you select the source of the software. It is not a good idea to install additional software during the initial installation of the operating system. If the software has a problem loading, this could cause problems loading or starting the operating system. Try to get the operating system installed and stable first, then install additional software.

*Select **None** and then left click Next >.*



Solaris 9 comes neatly packaged in what are known as Software Groups (also known as Software Clusters). There are five types of clusters:

Core Group An extremely limited number of software packages.

End User Group Contains the absolute minimum software needed to run the CDE (Common Desktop Environment) and network the workstation. This does not contain any documentation such as MAN pages. This is a good selection for a server or workstation that does not support desktop users. If a server is being built to just host Web pages or just act as an FTP server, this would be a good choice to use.

Developer Group This cluster has been designed for software developers. It has all the components of the End User Group and some additional programming libraries and development tools. This distribution also includes MAN pages and additional documentation. Most software developers request the Entire Solaris Software Group.

Entire Solaris Software Group

This cluster is the entire Solaris 9 Operating Environment. This software cluster contains everything in the Developer Group and the End User Group, plus additional Solaris components such as the FTP server, DNS server, etc.

Entire Solaris Software Group Plus OEM

This cluster is the entire Solaris 9 Operating Environment and additional device drivers for third-party equipment. This is a good choice if a Sun server has non-Sun components installed. For a typical workstation or server, this would be a bad choice because it loads the operating system down with extra drivers and software that will never be used.

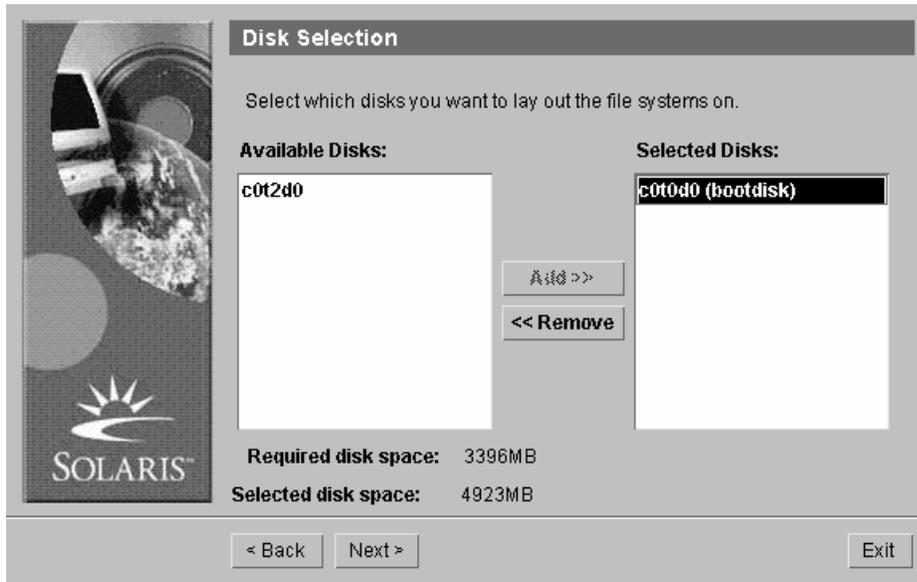
Select **Entire Group, Default Packages** and then left click Next >.

Just as a side comment, the system administrator can select **Custom Packages**, as shown above. This makes it possible to choose specific software packages out of the Group. This is somewhat dangerous because there could be software dependencies that you don't know about that will cause future problems. Use a lot of care when selecting this option. As mentioned before, the best option is to select is **Entire Group, Default Packages**. This has almost all the software tools and utilities needed for a server or workstation.

If your server or workstation is 100% Sun (no third-party devices) do not to use the Entire Group Plus OEM option. The Entire Group Plus OEM Software Cluster installs a lot of extra drivers and software onto the system that are not necessary for a 100% Sun server. Only select this option if your workstation or server has some type of special equipment that was not produced by Sun Microsystems.

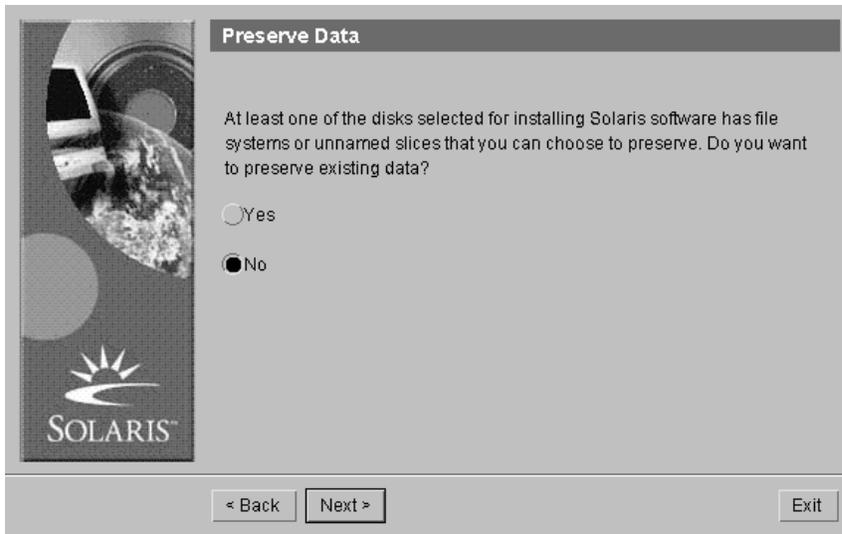
You will see the message:

The Solaris Installer is gathering cluster and package information
Please wait...



Next > *The disk naming convention is described in Chapter 13. For now, just use c0t0d0 as the installation disk. This is the first SCSI disk on the first SCSI controller. If the system has IDE disk drives it is the master disk on the first IDE controller. If the c0t0d0 disk is not found, use another hard drive that has at least 4 GB of disk space.*

*Do not select more than one hard drive for the installation. If you see more than one disk in the “Selected Disk:” column, click on the << Remove button until only one disk is shown. As mentioned above, select **c0t0d0** and then left click Next >.*



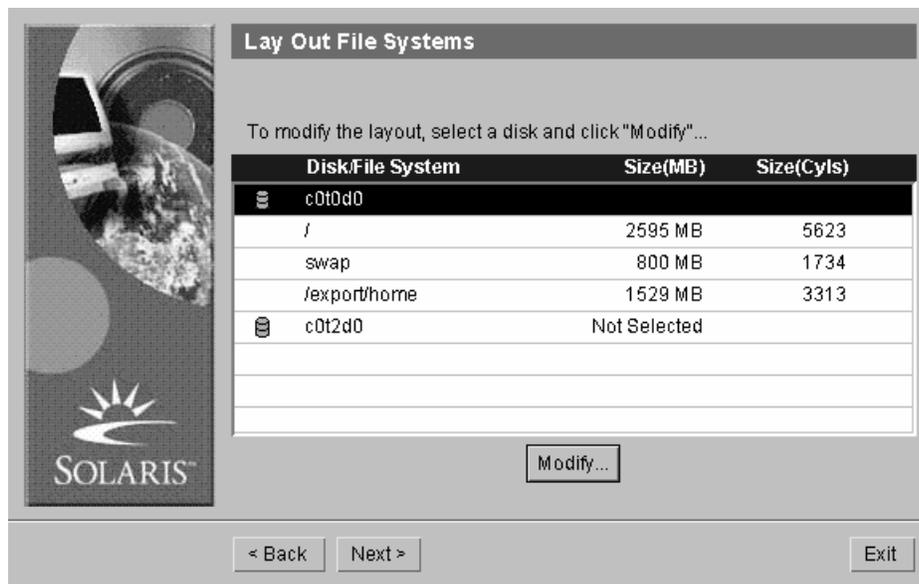
This option lets the system administrator preserve an existing file system and files found on the hard drive. However, it's best to have a fresh start.

As mentioned above, this will reformat the hard drive and erase all existing data!

*Select **No** and then left click Next >.*

The following message will now appear:

The Solaris Installer is determining size requirements for the packages selected



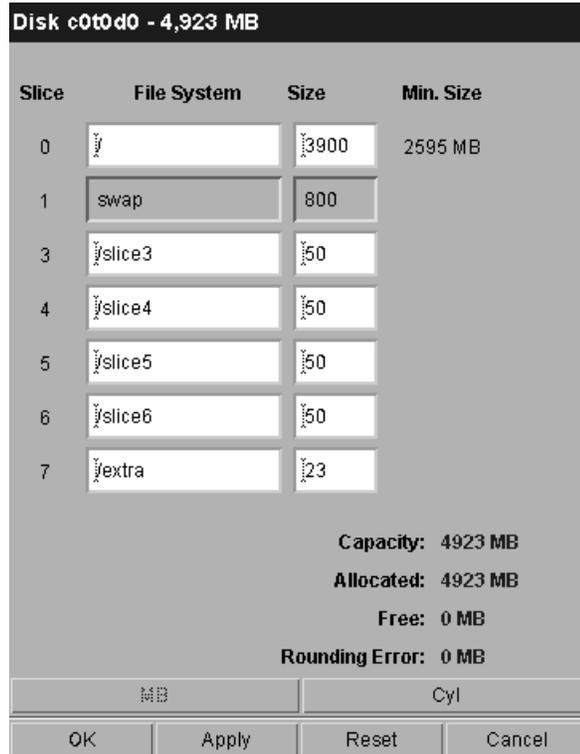
This screen lets you keep the default partitioning of your hard drive, or if the Modify... button is selected, you can change the partitioning of Solaris 9.

*The author of this book highly recommends that you select the Modify... button and don't use Sun's default partitioning. The problem comes from the fact that under the default partitioning, the **/export/home***

directory eats up all the disk space. This is ridiculous! When you want to install extra software, you may not be able to, because all the space is being used up by the /export/home directory. Be aware that after the installation, a disk can not be repartitioned.

For this lesson, select the c0t0d0 disk and left click on the Modify... button

The next figure shows some recommended slice sizes.



When the Modify... button is selected, this screen pops up. As can be seen above, the root partition was given a size of 3900 MB, /slice3, /slice4, /slice5 and /slice6 were given 50 MB. The last slice, the /extra slice took the remaining disk space, 23 MB. Slice 1, the swap partition, can not be modified from this screen. Slice 2 represents the entire disk and can not be modified here (and it is not even shown).

How you set up your disk slices depends on how large your hard disk is. Play with the slice sizes until there is no extra free space (the next to last line at the bottom right will read Free: 0 MB)

- If your system has a small hard drive (5 GB or less), try using the following settings:

Slice	File System	Size	Min.Size
0	[/]	[3900]	1594
1	[swap]	[can't change]	800
3	[/slice3]	[50]	
4	[/slice4]	[50]	
5	[/slice5]	[50]	
6	[/slice6]	[50]	
7	[/extra]	[*]	

* add any unused space in slice 7.

The smallest a slice should be is 50 MB. Anything less than 50 MB and the slice is almost completely useless. Play around with the size of the root partition (/) until it occupies most of the disk space. The number 3900 is not a firm number. If the hard drive just is not big enough for the selected software, remove the Solaris Documentation CDROMs and try again.

- If your system has a medium size hard drive (5 to 15 GB) try using the following settings. Play with the size of the root partition (slice 1 or /) until it occupies most of the hard drive's space.

Slice	File System	Size	Min.Size
0	[/]	[6000]	1594
1	[swap]	[800]	
3	[/slice3]	[1500]	
4	[/slice4]	[1500]	
5	[/slice5]	[1500]	
6	[/slice6]	[1500]	
7	[/extra]	[*]	

* use the remaining space in slice 7

- If your system has a very large hard drive (greater than 15 GB) try using the following settings:

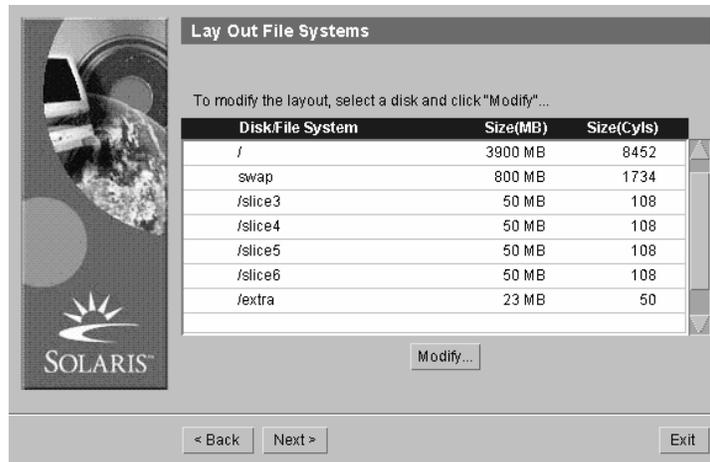
Slice	File System	Size	Min.Size
0	[/]	[8000]	1594
1	[swap]	[800]	
3	[/slice3]	[2000]	
4	[/slice4]	[2000]	
5	[/slice5]	[2000]	
6	[/slice6]	[2000]	
7	[/extra]	[*]	

* use any extra space in slice 7

From here you can adjust the size of the hard drive partitions. The root slice must have a minimum amount of disk space, depending on what software packages you selected during the installation. Each time you change an entry, the amount of free disk space is recalculated and shown on the bottom of the screen.

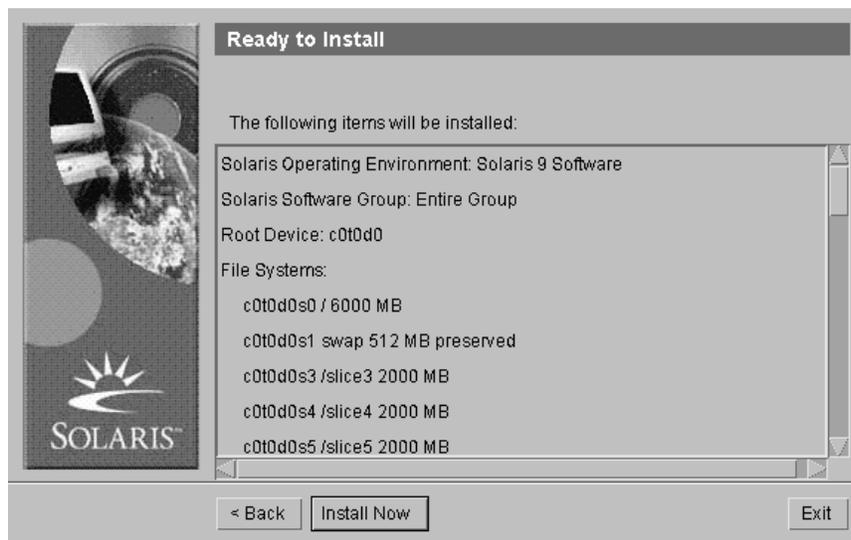
If the free space is negative, it will turn red. The installation program will not continue if you have tried to use more space than is available on the hard drive. In this example, slice 3, 4, 5, 6 and 7 are allocated under the /slice3 /slice4, /slice5, /slice6 and /slice7 directories. When the operating system is installed, the additional /slice# directories will be created. Each directory will have a small amount of space for storage.

After the OK button is selected, the final confirmation screen appears (the numbers shown here are for the suggested 5 GB configuration—the numbers you see may be different).



If you want to change any numbers, left click on Modify... and make the changes. When all the numbers are right, left click on Next >

The Ready to Install confirmation screen will now appear:



If there is a mistake, click on the < Back button until you come to the screen that deals with the incorrectly selected item. Continue doing this until you have fixed all mistakes in this screen.

Be aware that when you select Install Now, your hard drive will be erased and reformatted with the new operating system. There is no going back after this button is pushed. Do not do this if you still have critical data on your system that has not been backed up!

When everything is correct, and you are ready to install Solaris 9, click on the Install Now button.

The following message will appear:

Installing...

The following text appears as the Solaris 9 Operating System is installed. The names SUNWxxx are the names of the software packages.

Solaris 9 Software [Creating and checking file systems]
Solaris 9 Software [Installing SUNWctlu]
Solaris 9 Software [Installing SUNWcsu]

The install program will reboot the workstation and then immediately start the Web Start Install program. The install program will ask you to insert the second software CDROM.

Please specify the media from which you will install Solaris 9 Software 2 of 2 (SPARC Platform Edition).

Media:

- CD/DVD
- Network File System

To skip installation of Solaris 9 Software 2 of 2 (SPARC Platform Edition), click Skip.

Next > [Skip]

*Select **CD/DVD** and then left click Next >.*

The system now ejects the CDROM and prompts the user to insert the Solaris 9 Software 2 of 2 CDROM:

Please insert the CD/DVD for Solaris 9 Software 2 of 2 (SPARC Platform Edition).

Insert the Solaris 9 Software 2 of 2 disk.

WARNING

If the system does not eject the CDROM, open a terminal window (right click on unoccupied desktop space, left click on Tools, left click on Terminal).

After the terminal window opens, type the command "eject cdrom"

The CDROM will be ejected. Insert the Solaris 9 Software 2 of 2 CDROM. Then, type the command "volcheck".

The following messages appear:

Software loaded includes...

SunScreen 3.2

.

.

Checking System

This step verifies that all required Solaris Operating Environment Software has been loaded correctly.

Other messages include:

Launching Installer

Launching the Installer for the Solaris 9 Software 2 of 2 CD...

.

Ready to Install

The following items will be installed:

Product: Solaris 9 packages (part 2)

Location: /

Size: 535.7 MB

Solaris 9 package (part 2) 535.7 MB

< Back [Install Now] [Exit]

*Left click on the Install Now button.
The following message will appear:*

Installing Solaris 9 packages (part 2)

*Once the install starts, it cannot be stopped. There are no Exit or Cancel buttons to select.
When the installation is done, the following messages appear:*

As the install progress, various windows and messages will appear. As required, click on the appropriate button to advance to the next screen. The remaining windows and messages will vary, depending on the software that has been selected, and on the system architecture.

Installation Summary

The Solaris 9 Software 2 installation status is shown below:

Product	Status	Information
Solaris 9 packages (part 2)	Installed	[Details...]
.		
.		
.		
		[Exit]

Click on the "Details..." Button beside each product status for more information.

*The Details... button shows a small screen that has copy of the
/var/sadm/install/logs/Solaris_9_packages__part_2__install.B05251014 file.
This is one of the install log files created after a software installation.
This log file has information like*

Installing Package: SUNWxildh

**Copyright 2001 Sun Microsystems, Inc. All rights reserved.
8296 blocks**

Processing package instance <SUNWxildh> from </cdrom/Solaris_9/Product>

**XIL Loadable Pipeline Libraries
(sparc) 1.4.2,REV-2001.10.11
Using </a/usr> as the package base directory.**

Processing package information.
Processing system information.
4 package pathnames are already properly installed

Installing XIL Loadable Pipeline Libraries as <SUNWxildh>

##Installing part 1 of 1.

Installation of <SUNWxildh> was successful.

After the log file displays, other messages will appear:

Launching Installer please wait...

Pausing

Pausing for 30 seconds unless the Pause Button is pushed,

Reboot

Preparing to reboot...

Rebooting with command boot
Boot device: disk:a File and args:

*Notice that the boot command now uses the alias "boot" instead of the rather physical device tree /pci@1f,0/ide@d/disk@0,0:b That is because the install program modified the OpenBoot variable **boot** so that it points to the hard drive on which you installed the operating system. Also, note that the install program uses the :a argument after disk. The :a argument represents the first slice on the disk, or slice 0.*

SunOS Release 5.9 64 bit
Some rather generic boot messages.

Hostname: SUN100
Configuring /dev and /devices directory (compatibility devices)
The system is coming up. Please wait.
checking ufs file systems
/dev/rdisk/c0t0d0s7: is clean
RPC: Timed out
starting rpc services: rpcbind done
Setting netmasks of eri0 to 255.255.255.0
syslog service starting
Print services started.
sendmail [351] My unqualified host name (SUN100) unknown; sleeping for retry

The graphical login screen is called the "Login Manager. Now, the Login Manager starts, with the message

Welcome to <Hostname of the system>.

(In this case, the name is SUN100.)

There are 4 buttons on the bottom of the screen

OK Start Over Options Help

**At this point, DO NOT LOG IN AS THE ROOT USER !!!
FOLLOW LESSON 2.2 !!!**

Lesson 2.2 Creating a Generic User

This lesson shows how to create a generic user. The root user has full control over the Solaris 9 Operating Environment. But unfortunately, the root user is very dangerous, because it can easily destroy the Solaris 9 operating system. For safety's sake, most system administrators log in as a generic user and only use the root user when absolutely necessary

1. Left click on the Options button. Then hold down the left mouse button and select **Command Line Login**. A message should appear saying

```
*****  
*   Suspending Desktop Login...  
*  
*   If currently logged out, press [Enter] for a console login prompt  
*  
*   Desktop login will resume shortly after you exit console session.  
*  
*****
```

If no keys are typed, the console session will end and the Login Manager window will be displayed again.

2. Press the Return key for a console login prompt.
3. Type **root** and press the Return key (this is the root user's name).
4. Type the root user's password and press the Return key.

Figure 2.3 shows what the screen should look like when the root user enters a console prompt.

```
<HOSTNAME> console login: root  
Password:  
June 3 12:13:03 <HOSTNAME> login: ROOT LOGIN /dev/console  
Last Login: Wed June 2 08:13:23 from :0  
Sun Microsystems Inc. SunOS 5.9 May 2002  
#
```

Figure 2.3 Root User's Console Login

5. Now type these command in exactly as they are shown.

```
useradd -m -d /usr/user11 user11  
cp /usr/user11/local.profile /usr/user11/.profile  
passwd user11
```

6. At the **New password:** prompt, enter the password *< the password should be user11 >*
7. At the **Re-enter new password:** prompt, *< re-type the password user11 >*
8. For users with a null modem connection, type the following command
stty erase (then press the backspace key; it should produce the **^H** character)
9. If a CDROM or DVD is still in the CDROM drive, type the command **eject cdrom**

A SPARC-based system will not release a CDROM unless a proper eject command is given to release the CDROM or other media.

10. Type the keyword **exit** to end the session.
The following screen will appear:

```
*****  
*  
*   Resuming Desktop Login  
*  
*****
```

In the steps immediately above, a new user was created, with the login name user11.

Remember that it is dangerous for a novice user to play around with Solaris 9 as the root user. Because of this, in future lessons, always log in as user11 to learn the Solaris 9 operating system. User11 is a typical user in that the commands available under this login are restricted. User11 can cause some damage to the home directory assigned to that login, but user11 but can not damage the system.

When the CDE desktop starts again, log in as user 11. Type user11 as the user and press Return, then type user11's password and press Return. The CDE session will now begin. When the CDE desktop starts, three default screens pop up:

```
Help View  
File Manager  
Solaris User Registration
```

Notice that The Solaris User Registration window will continue to pop up until the user either registers Solaris 9 or chooses not to register it. To prevent this screen from showing up again, left click on the hyperlink "**Register**." Then when the screen appears again, click on **Cancel**. A third screen appears with the message "**Are you sure you don't want to register?**" Click on "**Never Register**" to prevent the Registration window from showing up again.

To close these screens, left double click two times quickly on the Window Menu button. This is the dash [-] symbol on the upper right hand corner of the screens.

To view the Help window again, left click once on the Help icon This Icon is shown in Figure 2.4



Figure 2.4 The Help Icon

Left click on the "Overview and Basic Desktop Skills" hyperlink. The next screen should have the title "Overview and Basic Desktop Skills" on the top. Left click on "Introducing the Desktop (located slightly below the title screen).

Shutting Down Solaris 9 Gracefully

Always be sure to shut down Solaris 9 correctly. Don't just press the power switch. This makes Solaris 9 experience a "hard crash." This means that the operating system did not perform its shutdown procedures, and the file systems were not synchronized when the system shut down. This can cause file system corruption and other

problems the next time Solaris 9 is started. Also, do not use the STOP + A keys. Instead, follow the next lesson to learn how to shut down the Solaris 9 operating system gracefully.

Lesson 2.3 Graceful Shutdown of Solaris 9

This lesson shows how to gracefully shut down Solaris 9 from within the CDE, and from the Console Login.

To shut down Solaris 9 from within the CDE

1. Log in as the root user.
2. Right click anywhere in unoccupied desktop space. The Workspace menu should appear, as shown in Figure 2.5.

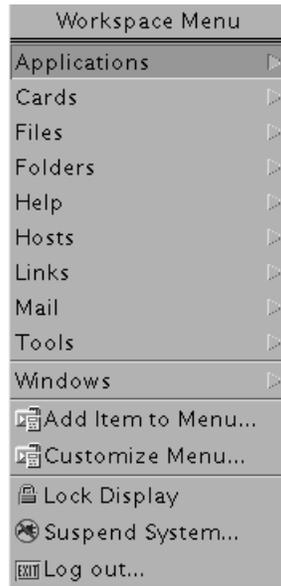


Figure 2.5 The Workspace Menu

3. Left click on the Tools menu item.
4. Left click on the Terminal icon.
The Terminal window will appear. It lets users enter commands directly into the system. The root user always has the pound (#) symbol as its login prompt on the left side of the screen.
5. Type the command **init 0**
*The command **init 0** is used to bring the operating system down gracefully.*

To shut down Solaris 9 from the Command Line Login

1. Left click on Options button. Then hold down the left mouse button and select **Command Line Login**. A message should appear, saying

```
*****
*                               Suspending Desktop Login...
*
*   If currently logged out, press [Enter] for a console login prompt
*
*   Desktop login will resume shortly after you exit console session.
```

*

If no keys are typed, the console session will end and the Login Manager window will be displayed again.

2. Press the Enter key for a console login prompt.
3. Type in the root user's name (**root**) and then press Enter.
4. Type in the root user's password (**root123**) and then press Enter.
5. Type the command **init 0**

An Additional Note

IMPORTANT: Solaris 9 has an option known as "Power Management." Under no circumstances should this option be selected. Don't use this option! If you do, software can become confused and the server or workstation can crash. Most seasoned system administrators avoid this option.

Quick Tip

- This tip is for senior level system administrators: Right click in the background area. A small Workspace menu will appear.
- Right click on Programs
- Select either the Terminal... or the Console... option.
- Now it is possible to run administration commands like **format**, **newfs**, etc. It is also possible to run GUI commands like **xclock** and other GUI tools not ordinarily available with the **boot cdrom -s** command.

Key Points to Remember

The Solaris 9 operating system is a fairly easy operating system to install. Almost every installation screen has some type of commentary and help screens. If for some reason Sun Microsystems changes the installation windows, just use common sense when installing Solaris 9. If in doubt, add extra software now, and then remove the software packages later on.

Chapter 3 The Common Desktop Environment

Lessons in This Chapter

Lesson 3.1 Using the Front Panel	3-5
Lesson 3.2 Using Workspace Buttons	3-6
Lesson 3.3 Renaming Workspace Buttons	3-7
Lesson 3.4 Adding Workspace Buttons	3-7
Lesson 3.5 Locking the Workstation	3-8
Lesson 3.6 Perform a Graceful Shutdown	3-11
Lesson 3.7 How to Suspend the System	3-13
Lesson 3.8 Using the Help Manager	3-15
Lesson 3.9 Installing Adobe Acrobat	3-17
Lesson 3.10 Working with the Floppy Drive (command line)	3-21
Lesson 3.11 Using the Find File Utility	3-23
Lesson 3.12 Using the File Properties Utility	3-24
Lesson 3.13 Calculator: Binary to Decimal Conversion	3-26
Lesson 3.14 Using the GUI Text Editor	3-29
Lesson 3.15 Using the Color Style Manager	3-31
Lesson 3.16 Create a Custom Icon	3-33

Introduction

The CDE (Common Desktop Environment) was originally created by the COSE (Common Open Software Environment) group. The members were Hewlett-Packard, IBM, Novell and Sun. Later, the original companies were joined by Digital, Hitachi and Fujitsu as sponsors of the CDE-Motif PST under the structure of the Open Software Foundation (OSF).

Solaris 9 comes pre-packaged with CDE (Common Desktop Environment) version 1.5. CDE lets the system administrator use various GUI (Graphical User Interface) based administration tools, such as the Admintool and Solaris Management Console. Previous version of Solaris had both the CDE desktop and the OpenWindows desktops. Sun Microsystems decided not to include the OpenWindows desktop with Solaris 9. Most companies do not install CDE on production-level servers because of the extra system overhead of running CDE. Also, most servers do not have a monitor or a graphics card installed.

On the other hand, although a desktop GUI interface does not have to be installed for Solaris 9 to work, most workstations do run CDE as a convenient display for novice Solaris users. There was some talk about Sun including the GNOME desktop with Solaris 9, but this did not occur fast enough for the initial Solaris 9 release. Sun does have an evaluation version of GNOME 2.0 in its download section at its <http://www.sun.com> webpage. The CDE desktop is the only GUI desktop with the Solaris 9 02/05 Media Kit. In the future it is expected that Sun will abandon the CDE desktop for the GNOME Desktop.

Before a user can access CDE, he or she must log in through the Login Manager. The Login Manager's screen should appear immediately after the operating system is started. The Login Manager lets the user select the language that CDE should start with (provided the language is installed for that version of CDE). The user can also choose to not start a CDE session and go directly into Command Line Mode. In Command Line Mode there is no GUI interface, just a text prompt. This is useful if the settings on CDE become damaged and CDE will not start

properly. Once a valid login name and password are provided, the Login Manager starts the Session Manager. The Session Manager starts CDE with the user's chosen language.

Once CDE is running, the user can open various GUI tools to check email, write text messages, perform basic system administration, and perform other common desktop tasks. Some common items found in CDE are the Front Panel, Application Windows, the Workspace menu, and Terminal Windows. These items make using CDE very easy to use.

The user can also customize CDE by choosing various fonts, background images and icon placements. CDE also has one very unique feature: the user can have multiple workspaces. Each workspace can be thought of as a separate desktop, with a separate name. One workspace could be labeled "servers" where various system administration tools like Process Manager, File Manager and Admintool are running. Another workspace could be labeled "mydesktop" where programs like Netscape Navigator, Calendar and an email program are running. A third desktop could be labeled "mycommunications" where an X-chat program is running. With the click of a button, the user can switch between different desktops for different tasks.

When the desktop user decides to leave CDE, his or her personal settings are saved by the Session Manager in the user's home directory. The Login Manager is then started again for a new session. When a user leaves CDE, the desktop settings are saved in his or her home directory (provided a home directory exists for this user). The Login Manager is displayed again, ready for a new CDE session. It is possible for a user to start another CDE session or start a command line session. In command line mode, there is no graphical interface displayed.

SPECIAL NOTE

If your Solaris 9 system does not show a GUI login manager when the operating system starts, read the "Administration of CDE" section at the end of this chapter. Most likely there is a resolution or display problem with the system or monitor. The Intel version of Solaris 8 was very picky about the graphics cards and monitors it would work with. Most Sun workstations and monitors are compatible with CDE and Login Manager. Check the Sun Solaris 8 HCL to check your video card and display settings.

The Login Manager

The Login Manager is fairly easy to use. Just type in a valid login name and password and then press the Enter key. The Login Manager lets the user select different options under the Options button. These include the location of the CDE source. Local CDE sessions are chosen by default. If a CDE server is located somewhere else (used with thin client architectures) select Options, then Remote Login. A list of remote CDE hosts is presented, and the user can select the host he or she wants to connect with.

The user can select a non-English language for CDE, provided that the CDE version for that language has been installed. Sun produces the CDE environment for many different countries and locales. The user can also choose to use a Command Line Session. A Command Line Session has only a text window with no graphics.

There is one other option, the Failsafe Session. This is used if a user accidentally does something that causes the CDE desktop not to start properly. The Failsafe Session only opens up a single Terminal window. From this window, command line utilities can be used to run simple commands to fix CDE. The Failsafe Session is also useful when a user just needs to type in a simple command and run it. After the Failsafe Session is over, the user just types **Exit** to leave the Failsafe Session.

If for some reason the Login Manager does not appear after the login name and password are entered, type the command `/etc/init.d/dtlogin start` to make the Login Manager appear. If it still does not work, read the section "Administration of CDE" for more advice. Other commands to try include the following three commands, in the order shown: `/usr/dt/bin/dtlogin -daemon;exit` followed by `/etc/init.d/dtlogin stop` followed by `/etc/init.d/dtlogin start` once more. To make sure that the Login Manager starts every time, type the command

```
/usr/dt/bin/dtconfig -e .
```

Once a valid username and password are entered, the Login Manager contacts the Session Manager. The Session Manager is responsible for saving and restoring applications, fonts, audio volume, mouse settings, and other user-customized features. The Session Manager looks for the user's previous desktop preferences in a directory named `.dt` and a file named `.dtprofile` in the user's home directory. If a user has never used CDE before, the Session Manager will create what is known as an Initial Session. The initial session is created from a system file named `/usr/dt/config/sys.dtprofile`. This file is copied to the `.dt` directory and `.dtprofile` file mentioned above.

The Session Manager will be covered in detail in the "Administration of CDE" section.

The CDE Desktop

After the Session Manager has started, the user is presented with a CDE desktop. When the CDE desktop starts, two windows will be open. The first window is the Help window. At any time, press the F1 key and the Help window will appear. Some applications capture the F1 key, so an application-specific help screen might appear instead of the CDE Help window. The Help window gives useful advice for novice users on how to use CDE.

The second window that appears when CDE starts is the File Manager window. The File Manager will start any time it detects a CDROM in the CDROM drive.

In some cases, a third window, called the Registration window, will also start. This window asks the user to register his or her copy of Solaris 9 with Sun Microsystems. Most system administrator disable the registration window because the company they work for has probably registered Solaris 9 in a bulk purchase.

Any of the CDE desktop windows can be closed by double left clicking on the dash [-] symbol at the top of the window.



Figure 3.1 The Front Panel

Using the Front Panel

On the bottom of the desktop is the Front Panel, shown in Figure 3.1. The Front Panel has two major components, the Main Panel and the subpanels (the Main Panel is shown in Figure 3.1).

The Main Panel has a center section that contains the Workspace buttons. These buttons are labeled "One", "Two", "Three" and "Four." When these buttons are pushed, a new desktop is presented to the user. Imagine if you had a computer that was capable of running four separate version of Windows XP. Now imagine you could switch between these computers with just a touch of a button. That is what the Workspace buttons allow you to do in Solaris. Actually workspaces are more complex than just another desktop but, for now just understand the Workspace buttons give you a new desktop.

On the left side of the Workspace button "One" is a tiny lock symbol. This lock symbol is used to lock the workstation. When the workstation is locked, only the original user's password or the root user's password can unlock the screen.

On the right side of the Workspace button "Four" is a small EXIT button. This button is used to end the CDE session. Depending on the configuration of CDE, a popup-window can appear, asking the user if he or she

really wants to exit the CDE session. Above the EXIT button is a small green globe. When this globe is left clicked, the user can enter a command line action.

On either side of the center section are two smaller wing panels that hold icons for some popular programs. Table 3.1 illustrates the icons that are displayed on the Front Panel.

Icon's Name	What it Look Like	Function
Netscape Navigator		Netscape browser, displays HTML pages
Calendar		Schedule appointments, set dates and meetings.
Home Folder		View the system's folders, move files, copy files, etc.
Text Note		Standard keyboard compatible text note creator, not a text editor.
Mail		Send and receive email messages
Printers		View printers
Desktop Controls		Change system font colors, background images, screen savers, etc. (personal settings)
CPU Disk Activity		Shows the activity of the CPU and the hard drive.
Help Manager		Onscreen help
Trash Can		Deleted files are saved here

Table 3.1 Icons on the Front Panel

Above some of these icons there is a triangular icon . When this icon is clicked, a subpanel menu is revealed. The subpanel rises up above the click point, revealing additional program icons. The triangular icon then points downward. If a user left clicks on the icon, the subpanel menu falls back down into the Front Panel. Figure 3.2 shows the subpanel that is revealed above the Netscape Navigator icon.

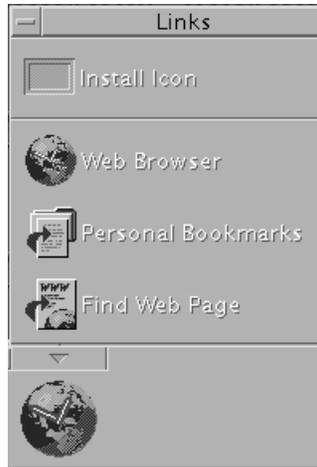


Figure 3. 2 Subpanel Above Netscape Navigator

On the far left and right edges are two gray vertical bars with a stripe pattern. These bars are known as “Move handles.”



When the mouse cursor is placed over the Move handles and the left mouse button is held down, the entire Front Panel display can be moved to a different location.

On the top left corner, above the left Move handle, is a small dot inside a square icon . If this icon is left clicked, the Front Panel display is shrunk down into an icon itself. When the icon is left double clicked it returns to its original size and position.

On the top right corner is a dash symbol inside a square icon . If this icon is left clicked, a standard CDE window menu is displayed.

Lesson 3.1 Using the Front Panel

In this lesson readers will practice using some of the icons and buttons on the Front Panel display. The first part of this lesson involves moving the Front Panel. Readers will then practice using the Lock icon. Later, readers will start one of the applications from a Front Panel Icon. Finally, readers will exit the CDE session.

1. Log on to CDE as the root user.
The root user's name is "root" by default. The password for the root user is whatever you chose when you installed the Solaris 9 Operating Environment. This should have been root123 according to the lesson.
2. Move the mouse cursor over the right Move handle.
The right Move handle is the vertical bar on the far right side of the Front Panel. It is just below the Minimize button (the Minimize button looks like a dot).
3. Hold down the left mouse button.
4. Drag the Front Panel anywhere on the desktop.



Try not to drag the Front Panel too far out of the desktop visible area, or it might be difficult to move it back to its original location.

- Practice dragging the Front Panel with the left Move handle.
The left Move handle is just below the Window menu button ((the Window Menu button is on the far left side. It looks like a minus sign -).
- Move the Front Panel back to its original location.
- Click on the Minimize button.
The Minimize button is located on the far right side of the Front Panel. It looks like a small dot (.) inside a box. At this point the Front Panel should disappear and a small icon should represent the Front Panel. The name of the Front Panel icon is the name of the workspace that is open. In this case it should be the word "one."
- Left double click on the Front Panel icon to enlarge it again.



- Click on the triangle above the Mail icon.
The Mail icon looks like an office inbox. A subpanel should rise up above the Front Panel.
- Double left click on the subpanel's Window menu button.
The Window menu button looks like a minus (-) sign and is located on the top left of the subpanel. This should lower the sub panel back down.



- Click on the Text Editor icon.
It looks like a pencil over a piece of paper. In this case it is set up to save a not. The note can also be saved to a text file.
- Play with the Text Editor, but don't save a file anywhere on the system.
- To close the Text Editor, double left click on the Window menu button.
The Window menu button looks like a minus (-) sign and is located on the top left of the window.
- Left Click on the EXIT button.
The EXIT button is slightly to the left of the Workspace button labeled "Four" (in the center of the Front Panel).



Workspace Buttons

In the center of the Front panel are the Workspace buttons. These buttons are labeled "One", "Two", "Three" and "Four."



When a user clicks on one of these buttons, a new desktop is presented. By default, the first workspace that is used is associated with the "One" button. If a user double clicks on the "Two" workspace button, a new desktop is created.

Lesson 3.2 Using Workspace Buttons

In this lesson readers are shown how the different workspaces work in CDE. The reader will create a different background for each workspace.

- Log in as the root user.
The root user's name is "root" by default. The password for the root user is whatever you chose when you installed the Solaris 9 Operating Environment. This should be root123 according to the lesson.
- Left click on the "Two" button in the center of the Front Panel.

You are in the second workspace. Just as a quick reference, a Sun mouse has three buttons, The left button is 1, the middle button is 2, and the right button is button 3 (left to right, 1, 2, 3). Some instruction manuals will say click button 1, which means to click the left mouse button.

3. Left click on the "One" button.
At this point you should be in the first workspace again.



4. Left click on the [EXIT] icon. 
The Login Manager should be displayed again.

Renaming Workspace Buttons

The Workspace buttons can be re-labeled to something more meaningful than "One," "Two," "Three," or "Four." Select names like "MyDesktop" or "Seattle" or "Xserver32" or whatever. To change the names of a button, slowly left double click on it. The first click will highlight the button, and the second click will open a text window over it. Rename the button to something more meaningful than One or Two. After the button has been renamed, press the Return key.

Lesson 3.3 Renaming Workspace Buttons

In this lesson the reader will rename the first workspace button from "One" to "mydesktop". The Workspace buttons should be renamed so they have more meaning than "One" or "Two."

1. Log in as the root user.
2. Left click on the Workspace button labeled "One" only once.
3. Left click on the Workspace button labeled "One" again.
The button should now have a white background with black lettering.
4. Press the BACKSPACE key several times.
5. Type in "mydesktop" as the new name.
6. Press the Return key.
The button's name should read "mydesktop"



Most seasoned administrators organize the workspaces by the type of work that is being performed in a desktop. For example, the first workspace could be labeled "servers." This desktop would have the Admintool and Process Manager windows running for each server that the administrator was responsible for. The second workspace could be labeled "mydesktop" and contain the administrator's favorite email program, the X-chat program and other programs associated with his or her workstation. The third workspace could be labeled "programming" and have two or three C programs running.

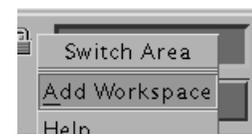
The important thing to understand is that each workstation should be organized by tasks. The worst situation occurs when each workspace has an email program running and a text file in each of two windows and a program being debugged in three windows. After a short time there is a spattering of different applications running all over the place. Then the beauty of multiple workspaces becomes a nightmare, where the user does not know what application is running in what workspaces! Keep your workspaces organized!

It is possible to add extra workspace buttons onto the Front Panel.

Lesson 3.4 Adding Workspace Buttons

In this lesson the reader will add an extra workspace button onto the Front Panel.

1. Log in as the root user,
2. Left click on the Front Panel, slightly to the left of the Workspace button labeled "Three."
A popup menu should appear from where the mouse button was clicked.
3. Left click on the Add workspace menu item.



4. A new Workspace button should appear, labeled "New."

5. Right click on the "New" button

A popup menu should appear over the button.

6. Select Delete from the popup menu

The new button will be removed with the Delete menu item.

7. Left click on the Desktop Controls icon.

8. Left click on the "Maximize" button on the new window.

The Maximize button is on the far right top corner of the window. It is a square inside a square. There are several applications in this window that can be used to customize the desktop. Play with these applications.

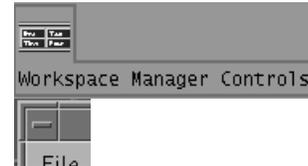
9. Left double click on the Workspace Manager Controls icon.

Play with the Workspace Manager Controls. Add extra buttons, etc.

10. Left double click on the Window Menu buttons on all the displayed windows.

These buttons look like a dash (-) symbol on the top left corner of the window. Double clicking these buttons will kill the window.

11. Exit the CDE session by left clicking on the EXIT button.



Try to limit yourself to six or eight Workspace buttons on the Front Panel. Any more than that, and the Front Panel starts to become rather cumbersome to work with on a standard size monitor.

If after removing or adding some Workspace buttons, the Front panel looks odd, just log off from CDE and log back on again. The Front Panel will be resize itself automatically when the session manager starts CDE again.

Locking the Workstation

One of the easiest ways for someone to hack into a system is to use an unattended console. This could let that person add an additional root user for later malicious hacking. Never leave a workstation or server alone without having the screen locked. If there is a hacking incident, the only thing the security experts will report is that the hacking came from your workstation with your login name during business hours.

On the Front Panel there is an icon that looks like Figure 3.3.



Figure 3.3 The Lock Icon

The Lock icon prevents the keyboard and mouse from working. It will also start the screen saver if one has been selected by the user. Only the password from the user that locked the screen, or the root user's password, can be used to unlock the system. The reason that the root user's password can also unlock the screen is because novice users have a bad habit of forgetting their passwords.

Locking a workstation or server does not harm any programs that are running. If an email program is running or a C program is running in the background, these programs are not harmed by the screen lock.

Lesson 3.5 Locking the Workstation

In this lesson the reader will lock the CDE desktop and then unlock it.

1. Log in as the root user

2. Click on the Lock icon located slightly left of the center of the workspace.

3. Type the root user's password to unlock the display.

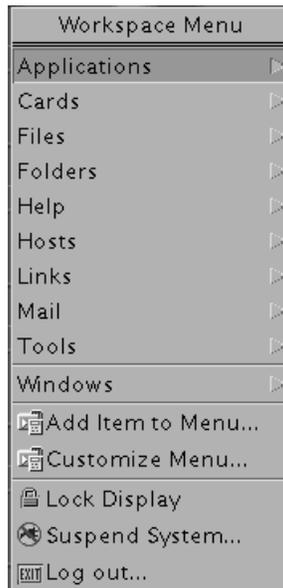
4. Left click on the EXIT button on the Front Panel display.

5. Log in as the root user again.

6. Right click anywhere in unoccupied desktop space.



7. Left click on the Lock Display icon on the bottom of the window.
8. Right click anywhere in unoccupied desktop space. This will display the Workspace menu.



9. Left click on the “Log out...” icon on the bottom of the Workspace menu.

Starting a Command Line Session

A user can start a Command Line session from the Login Manager. The Command Line Session will only display text on the screen. The X server (which allows the CDE to run) is not running once a Command Line session starts.

To start a Command Line session, left click on the Options button. Select the “Command Line Login” option. Type in your username and password. If the root user’s account is used, a pound (#) symbol is shown on the left of the screen. To exist from a Command Line Session, type the word “exit” and the Login Manager will start again.

The Command Line mode is very useful in cases where the CDE desktop and the graphics have become badly damaged for some reason. There are command line utilities that can be used to repair the X server configuration files used to control the graphics of CDE. Also, there are times when a workstation or server should not have CDE running due to security concerns or system resource issues. If a server has a runaway program or process and can just barely function, trying to work within CDE can become almost impossible, because it takes so long for simple things like screens to refresh and windows to open or close. A Command Line session requires almost no system resources to speak of.

Also, if CDE on a server or workstation is damaged, there is a chance the system will not shut down properly and will hang while trying to turn off the server. The safest way to handle a server with a damaged CDE is to enter a Command Line session and then shut down the server. If CDE freezes during the shutdown process, you won’t be able to enter a Command Line session. If this happens, connect to Serial Port A and establish a terminal server session. Once the terminal server session has been established, shut down the server with the command **init 0**.

The Workspace Menu

The Front Panel displays some icons, but it is rather limited in what it can display. If a user right clicks anywhere in unoccupied desktop space, a popup menu appears that gives the user the ability to choose from some

more applications. This popup menu appears wherever the mouse was last right clicked on the desktop. Figure 3.4 is a screen shot of the Workspace menu.

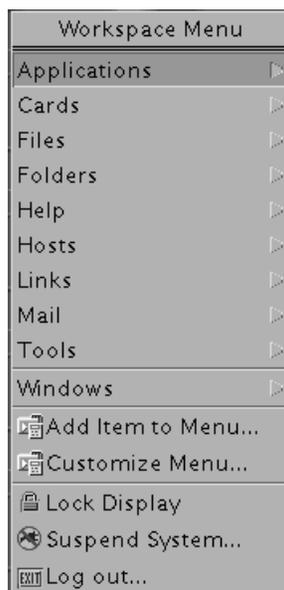


Figure 3. 4 The Workspace Menu

Shut Down Solaris 9 Gracefully

The Solaris 9 Operating Environment must be shutdown in an orderly manner from a shutdown command. Pressing the power switch will most likely result in the file system becoming damaged and the operating system having difficulty starting again. Only the root user can shut down the server. There is one exception, though: if a user has been granted the privilege of shutting down the server with RBAC (Role Based Access Control). RBAC is covered in the Advanced section of this book. For now, know that a typical user can not shut down Solaris 9.

There are several commands that can be used to shutdown Solaris 9. These commands can be run from within a terminal window within CDE, a console window from within CDE or from the terminal session connected to Serial Port A. These commands will run from anyplace where the root user can type in command.

The `init` command

The `init` command is a very direct command that is not difficult to use. Just type the command `init 0` to bring the system down to the `OK` prompt. The `init` command tells Solaris 9 to enter into a given run level. Run levels are covered later in the book. For now, know that run level 0 makes Solaris run through all the proper shutdown routines and scripts. Run level 5 tells Solaris 9 to shut down gracefully, just like run level 0. The big difference is that with run level 5, Solaris 9 also tells the SPARC hardware to power off after the operating system has completed its shutdown tasks. This can be useful when a server is in a remote location. Unfortunately, there is no remote way to start a SPARC server.

At the `OK` prompt, the power can be safely turned off, or the command `power-off` can be typed on systems that have the capability of shutting themselves down. The command `init 5` can be used to halt the operating system and automatically power off the SPARC system once the operating system has finished its shutdown routine.

The **init** command is in the **/sbin** directory and has the absolute path name of **/sbin/init**. For example, the command **/sbin/init 5** can be used to shut down the server if the **/sbin** directory is not in the **\$PATH** variable.

Lesson 3.6 Perform a Graceful Shutdown

This lesson starts with a CDE session. The reader then opens a Terminal window. The command **init 5** or **init 0** is used to shut down the system. When the system is powered back on, the reader will start a Command Line session instead of a CDE session. From the Command Line session the reader will type the command **init 0** at the root user's prompt. The system (if it is a SPARC-based system) will come down to the **OK** prompt. From here either the power switch can be pressed or the command **power-off** can be typed at the **OK** prompt.

1. Log in to CDE as the root user.
Remember that the root user's login name is root and the password is what was set up when Solaris 9 was set up in the beginning.
2. Right click anywhere in unoccupied desktop space.
The Workspace menu should appear
3. Left click on the Tools menu choice.
4. Left click on the Terminal icon.
A terminal window should now be opened on the desktop.
5. Type the command **init 0**
6. At the **OK** prompt, type the command **power-off**
*The command **power-off** tells the OpenBoot firmware to shut down the server. For older systems that do not support the **power-off** command, press the power button to turn off the system's power.*
7. Press the Power On button on the workstation.
*If the Solaris 9 operating system does not start and the system stops at the **OK** prompt, type the command **boot**. This will start the Solaris 9 operating system. If the **boot** command does not work, try the command **boot disk**.*
8. When the Login Manager is displayed, left click on the Options ▾ button.
9. Left click on "Start Command Line Session"
There should be a message on the screen telling you that you can leave the Command Line session by typing the word "exit"
10. Log in as the root user.
11. At the root user's prompt (the pound sign #) type the command **init 5**
*The command **init 5** gracefully shuts down the operating system and powers down the system. Older SPARC systems might not be able to work with the command **init 5**. In that case, type the command **init 0** and then press the power button for 5 seconds. Intel-based systems definitely will not know what the command **init 5** represents, so just type **init 0**. The server should be powered off at this time. Press the power button again to restart the system.*

The shutdown Command

The **shutdown** command works just like the **init** command, except that it can only shut down a server or reboot a server. It can not work on some other run levels. The **shutdown** command is very useful for a server or workstation that has several users still actively logged on, because it sends out a warning message to all users.

```
shutdown -g120 -i0 -y "server going down"
```

The "120" after "-g" means that the system will shut down in 120 seconds. The following message then appears on every user's console:

```
Broadcast Message from root (console) on sun100 Sat Sep 14 17:17:54...  
The system sun100 will be shut down in 2 minutes  
server going down
```

1 minute later, every user will see the message

```
Broadcast Message from root (console) on sun100 Sat Sep 14 17:18:54..  
The system sun100 will be shut down in 1 minute  
server going down
```

30 seconds later, every user will see the message

```
Broadcast Message from root (console) on sun100 Sat Sep 14 17:19:24...  
The system sun100 will be shut down in 30 seconds  
server going down
```

When the server goes down, every user will see the message

```
Broadcast Message from root (console) on sun100 Sat Sep 14 17:19:54...  
THE SYSTEM sun100 IS BEING SHUT DOWN NOW ! ! !  
Log off now or risk your files being damaged  
server going down  
Changing to init state 0 - please wait.
```

After the final message, the server shuts down.

These messages will appear on almost any terminal session, telnet session, console screen, exported CDE screen or any other text screen that users can see. The **shutdown** command has by default a 60 second delay from the time the command is issued until the time the system actually starts its shutdown routine. As noted above, the time delay can be changed with the **-g** option. The shutdown command can have a customized message like "Everyone, I am taking the server down for 10 minutes" instead of the standard shutdown messages.

The **shutdown** command supports the following options:

```
shutdown < options > <run_level> custom_message
```

< options >

-g <seconds> Specifies the grace period that the server waits for users to log off before the server starts to go down. The default is 60 seconds. After the **-g** option, type in the number of seconds for the grace period.

-i <run level> The run level Solaris 9 should enter. Run level 0 is complete shutdown, run level 5 is system shutdown and power off. Other run levels will be discussed in later chapters in the book.

-y Without the **-y** option selected, Solaris 9 asks the question "Do you want to continue? (y or n)" The **-y** option gives an automatic yes to this question.

custom_message Any text message.

The halt, reboot and poweroff Commands

The **halt** command is used when a server must go through an emergency shutdown. This command does not give out a warning message like the **shutdown** command. Also, any running process or program is given a

hard kill signal. The processes are not allowed to gracefully shut themselves down, and this can result in file corruption and program errors.

Do not use the `halt` command except in emergencies. Solaris 9 uses start and kill scripts to gracefully start and end programs in the `/etc/rc#.d` directories. The `halt` command also does not run the proper shutdown scripts in the `/etc/rc#.d` directories.

The `reboot` command powers down the system and returns it to multi-user mode or run level 3. The `reboot` command can only be run when the server is in single-user mode.

The `poweroff` command does the same thing as `init 5`. This command lets the Solaris 9 Operating Environment shut down gracefully and then shuts down SPARC-based systems.

How to Suspend the System

Solaris 9 supports what is known the “Suspend System” utility. Basically, all of the system information and memory is stored. The `boot-device` variable is reset to the suspended image. The SPARC system then shuts down and powers off.

This is a useful utility for a workstation, but it should never be used on a production level server. There are some applications that have time stamps and other time dependent functions. If the application suddenly “wakens” to a new system time, it could cause problems for the application.

Lesson 3.7 How to Suspend the System

In this lesson the reader will learn how to suspend and revive a system. This should only be done on a workstation, never on a production server.

1. Log in as the root user.
2. Right click anywhere in unoccupied desktop space.
3. Left click on the Suspend System... menu choice at the bottom of the screen.

A message appears that says

Saving System State. Please wait...

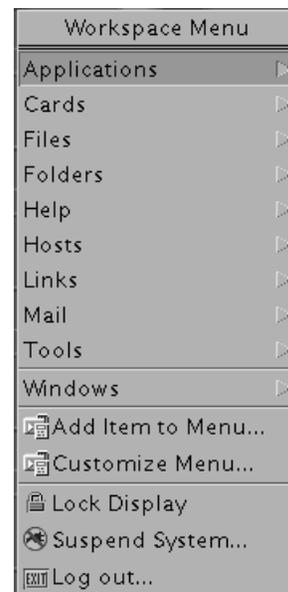
The system should power itself off automatically

Power on the system

The boot device has been changed to the swap partition

Restoring the System, Please wait...

4. The system comes back up, but the CDE is locked.
5. Type in the root user’s password to unlock the system.



CDE Applications

This next section describes some very common CDE applications. Most of these applications can be used by both the root user and ordinary users.

Help Manager

The Help Manager is the icon on the Front Panel that looks like a set of books with a question mark on them. This application basically helps users find help topics on the CDE. The Help Manager icon pulls up the Help Viewer window. This window features the Welcome to Help Manager window. Figure 3.5 is a screen shot of the Help View window.



Figure 3.5 The Help Viewer Window

On the top of the window is the Menu. There are four menu items that can be selected.

<u>F</u>ile	This menu item has three submenu items.
<u>P</u> rint...	Used to print the help topic on a printer.
New <u>W</u> indow...	Copies the current window to a new window.
<u>C</u> lose	Close the window.
<u>E</u>dit	
<u>C</u> opy	Copies text from the help menu.

<u>S</u>earch	<u>I</u> ndex	Opens an Index Search window used to do word searches on all the volumes or selected volumes.
<u>N</u>avigate	<u>B</u> acktrack	Return to the prior window, like the Back button on most browsers.
	Home <u>T</u> opic	Return to the starting help page, like the Home button on most browsers.
<u>H</u>elp	<u>O</u> verview	Overview of using the help system.
	Table of <u>C</u> ontents	Help-based Contents menu.
	<u>T</u> asks	Help discussion based on tasks.
	Reference	Help references.
	<u>M</u> ouse and Keyboard	How to use the mouse and keyboard.

On the right side of the Help View are four very useful buttons

Backtrack	Returns the user to the previous help page.
History...	Shows previously viewed help pages.
Index....	Opens an Index Search window used to do word searches on all the volumes or selected volumes.
Top Level	Returns to the home page. This page displays all the product families that have been installed and registered with the Help Manager. Depending on the installation of documentation, there should be two product families by default.

The following topics are at the upper left of the Help Manager window, below **Welcome to Help Manager**:

Common Desktop Environment	Has all the on-system documentation for the CDE.
Overview and Basic Desktop Skills	A very low level help menu that describes how to use the mouse and keyboard.

Lesson 3.8 Using the Help Manager

In this lesson, readers will use the Help Manager window to examine the Common Desktop Environment documentation. This documentation describes how to navigate around the CDE.

1. Log in as user11 (this user was created in Chapter 2).
 2. Left click on the Help icon located on the right side of the Front Panel window.
Figure 3.5 shows what the Help window looks like.
 3. Left click on the hyperlink "Common Desktop Environment."
 4. On the next screen that appears, left click on the hyperlink "Introducing the Desktop."
 5. Left click on Backtrack button on the upper right side of the window.
 6. Left click on History...
 7. Left click on Close.
 8. Left click on Index...
- The Index... button lets the user perform a test search through the help volumes.*
9. Left click on "All Volumes "



10. Left click on Entries with: **mouse**.
11. Left click on Start Search.
12. Left click on Close. Close the Help - Index Search window.
13. Left click on the Top Level button.
The Top Level button returns the user to the top of the Help webpages.

Solaris Documentation

Sun Solaris in the past used what was known as an AnswerBook2 server. Solaris 9 has shifted to using HTML webpages and Adobe Acrobat **pdf** files. For the webpages, all a company needs is an HTML 3.2 compliant web browser. To view Adobe Acrobat **pdf** files, an Adobe Acrobat reader must be installed on the system.

There are two ways to install Solaris 9 Documentation, the Installer utility and the **pkgadd** command. To install the documentation with the Installer utility, insert the Solaris 9 Documentation CDROM in the CDROM drive. The File Manager should now start automatically. If the File Manager does not start automatically, right click anywhere in unoccupied desktop space and bring up a popup menu known as the Workspace menu. Left click on Files, then left click on File Manager. (There is also a File Manager icon on the Front Panel window that looks like a filing cabinet with papers and folders hanging out of the top.)

Once the File Manger has started, navigate to the CDROM. There should be an icon for the installer. It looks like a small MAC computer. Left double click on the installer icon. A small popup window will first come up. Don't close this window! A Solaris 9 Installer Welcome window will now appear. It should have the title "Installing Solaris 9 Documentation." Click on the Next> button. From here it is possible to select either HTML or PDF documentation. Make sure to select the Custom install options. If the default install is selected, documentation written in Dutch, Spanish, French, Italian and Swedish will be installed on the system, wasting valuable hard drive space.

The documentation is installed under the **/opt** directory. To install languages other than English, select the following languages with the abbreviations shown below:

de	- Dutch
es	- Spanish
fr	- French
it	- Italian
sv	- Swedish

For systems with severe shortages of disk space, select only Adobe Acrobat documentation. That will only occupy 83.7 MB of disk space. If the HTML documentation is selected, it will take up 107.1 MB of disk space. If both the HTML and Adobe Acrobat documents are installed, the disk space required climbs to 190.7 megabytes. If everything is installed (the default), the documentation takes 332.9 MB.

The HTML documents are located in the **/opt/sun_docs** directory. To look at this documentation, open the File Manager and double left click on the sundocs.html webpage. This webpage is the home page for all the other HTML documentation webpages.

To install the Solaris 9 documentation from a command line, open a terminal window. To open a terminal window, right click anywhere in unoccupied desktop space. Release the mouse button, then left click on Tools and then left click on Terminal. Once in the terminal window, type the commands:

```
cd /cdrom/cdrom0/Solaris_9_Doc/common  
pkgadd -d .
```

Select the number that corresponds to the document that should be installed.

Obtaining Adobe Acrobat Reader For Solaris

To view the documentation in Adobe Acrobat format, it is a good idea to install the Adobe Acrobat 5.0.5 reader for Solaris. It's always a good idea to load the latest version of Adobe Acrobat onto a new Solaris installation, just to make sure the Acrobat reader is current.

The Adobe Acrobat reader can be downloaded from <http://www.adobe.com>. From there, click on Support, Download Acrobat Reader. Scroll down the webpage and select Get Acrobat Reader (this is actually Acrobat 5.0.5 with security). Select Sun Solaris SPARC. For systems that have Solaris 8 for Intel, Select Solaris Intel version. A file named **solaris-505.tar.gz** is copied to the hard drive. This is a 10 MB file, so it could take quite some time to download it by modem.

To un-compact this file, type the commands:

```
gunzip solaris-505.tar.gz
tar -xvf solaris-505.tar
```

If the Solaris system is not directly connected to the Internet, follow the steps in the following lesson.

Lesson 3.9 Installing Adobe Acrobat

This lesson assumes that the user has a CD burner on their PC. The Adobe Acrobat file (**solaris-505.tar.gz**) can also be copied from another machine via FTP if the user knows how to use FTP.

1. Rename the file **solaris-505.tar.gz** to **myadobe**
This avoids possible problems with software that truncates long filenames into old MS-DOS 8.3 filenames.
2. Burn the file myadobe onto a regular CDROM
It does not matter what platform the CDROM burner is located on.
3. Insert the CDROM into the Solaris 9 workstation (*If there is already a CDROM in the drive, first type the command **eject cdrom** ..*)
4. Type the command **mkdir /acrobatreader**
This command makes a directory named /acrobatreader
5. Type the command **cp /cdrom/cdrom0/* /acrobatreader**
This copies the myadobe file from the CDROM. The File Manger utility is also a viable way to copy the file.
6. Rename the file back to its original name.
The command is:
mv /acrobatreader/myadobe /acrobatreader/Solaris-505.tar.gz
7. Type the command **cd /acrobatreader**
8. Now type the commands:
gunzip solaris-505.tar.gz
tar -xvf solaris-505 *.tar
9. Now type the command **./INSTALL** or **/acrobatreader/INSTALL**
Notice that there is a dot slash (./) in the front of INSTALL. The ./ tells Solaris to look in the current directory for the command.
10. Press the space bar until the legal message is through, and type **accept**
11. Let Adobe install the software in the **/opt/Acrobat5** directory.
12. To start the Adobe Acrobat reader, type the command:
/opt/Acrobat5/bin/acroread
If you are familiar with the File Manager Utility, copy the acroread icon onto the desktop.
13. Type the command **rm -rf /acrobatreader**
*The command **rm -rf** deletes a directory and all sub-directories and files in the directory.*

File Manager

The File Manager is a GUI utility designed to make working with files, applications and folders very easy. The File Manager utility shows files, applications and folders with different icons. Figure 3.6 is a screen shot of the File Manager.

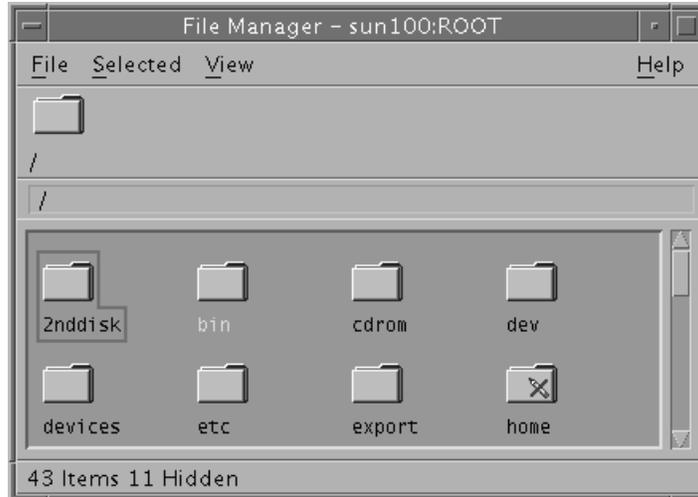


Figure 3.6 File Manager Icons (Application, File, Folder)

These icons are shown in a view window at the bottom of the File Manager.

The File Manager has an icon on the Front Panel that looks like a filing cabinet with some papers and folders hanging out of the top. Figure 3.7 shows what the File Manager icon look like.



Figure 3.7 The File Manager Icon

Basic Navigation within the File Manager

To move down into a folder, left double click on it. To move back up the directory tree, left double click on



the icon labeled "**.. (go up)**". This icon always appears on the upper left part of the view window. The only time the "go up" icon does not exist is when you are at the root (/) directory. There are no folders higher up than the root directory.

To do anything in the File Manager, use the left mouse button to select objects. You can choose multiple objects by holding down the left mouse button and drawing a box around them. Once objects are selected, left click on **Select**ed and then on the menu item that performs the function needed.

The **Select**ed menu has the following submenus. They are available when only one object has been selected. If more than one object has been selected, they are grayed out and are unavailable:

Move to... Move the file, folder or application to a new location.

- Copy to...** Copy the file, folder or application to a new location.
- Copy As Link** Creates a Symbolic link to the file. Only the file's name is copied, not its contents. The file's contents stay in the same place. This is similar to using Microsoft Windows shortcut icons.
- Rename...** Rename the file, folder or application to a new name.

As mentioned above, when multiple items are selected, the Move to..., Copy to..., Copy As Link and Rename menu selections are grayed out. The menu items that do work if multiple items are selected are:

- Put in Workspace** Puts the contents in a temporary workspace item. These items can then be dragged to a new location.
- Put in Trash** Put the items into the trash can.
- Properties...** Reset the properties on the selected items.
- Select All** Select everything visible in the window
- Deselect All** Remove everything visible in the window.

Creating a New Folder

To create a new folder, click on File , then on New Folder... The popup-window that comes up is fairly intuitive to use. Just understand that Solaris 9 uses the front slash character (/) instead of the backslash character (\) to denote a directory in Microsoft Windows. For example, a new folder can be named /**myfolder** but not \b**myfolder**. Also, understand that drive letters like **c:** and **d:** do not exist in Solaris 9.

Deleting a Folder

To delete a folder, just left click on the folder and then choose **Selecte**d**, Put in Trash**.

Using File Manager on CDROMs and DVDs

The File Manager is an excellent way to work with removable media like a floppy disk or a CDROM. File Manager starts any time CDE starts and a CDROM is in the CDROM drive. File Manager also displays itself whenever a new CDROM is inserted. You can use File Manager to eject a CDROM from the CDROM drive. To do this, select **File** from the Menu bar, then select Eject. You can not just push the eject button on the front of the CDROM drive to eject a CDROM. All programs and users that are accessing the CDROM have to be disengaged from the CDROM before it can be ejected.

Using File Manager with Floppies

File Manager has a menu option on the File Toolbar menu that allows a user to check a floppy drive. Understand that Solaris can not automatically detect the insertion or removal of a floppy disk. It's different with CDROMs because the CDROM drive sends a signal out when its door is opened or closed. A floppy drive does not have any way of detecting when a floppy is inserted or removed.

To tell File Manager that a new floppy has been inserted into the workstation, click on File, then Open Floppy. This starts File Manager. Then, click on File and then select Open Floppy.

Solaris 9 recognizes four different file formats:

PCFS (DOS)	This works with MS-DOS based file systems.
UFS (UNIX)	Universal File System, works with UNIX and UNIX-compatible systems.
UDFS	Universal Disk Format System, works with a variety of operating systems.
NEC DOS	A DOS file system used with NEC systems.

How to Format a Floppy Disk with File Manager

Be careful when formatting a floppy disk. The latest version that the author has seen of File Manager's online help page on this topic is incorrect. To format a floppy disk with the File Manger, first insert an unformatted disk in the floppy drive. Left Click on the Toolbar Menu option File , then click on Open Floppy. If the floppy drive is not formatted, a format window will appear that will give the user the ability to format a floppy disk. Figure 3.8 shows the Format Floppy window.

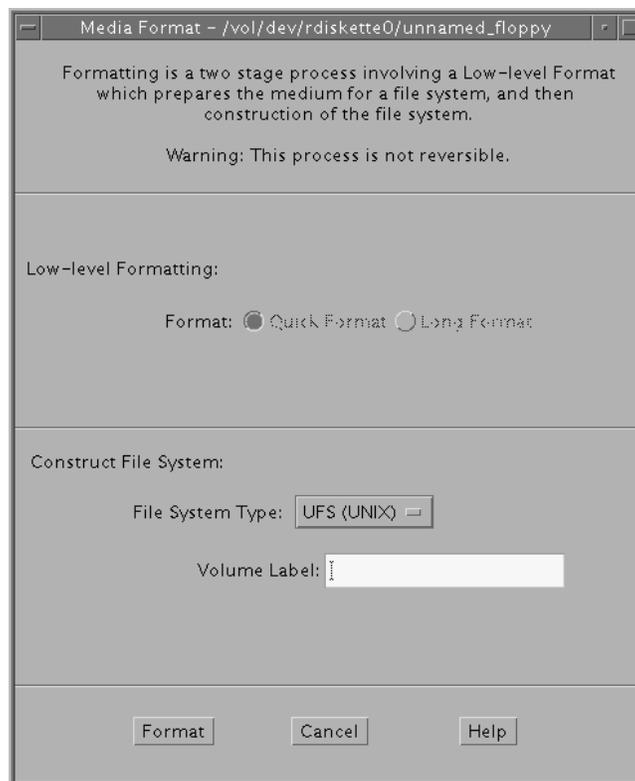


Figure 3.8 Format Floppy Window

To reformat a disk that is already formatted, the user will have to open the Format window manually. To do this with the File Manager, first insert the formatted disk in the floppy drive. Left click on the Toolbar Menu option File, then left click on Open Floppy. A new File Manager window will appear, showing the contents of the floppy. Left click on the Toolbar Menu option File , then click on Removable Media Manager. The Removable Media Manager will now show an icon of the disk., as shown in Figure 3.9.

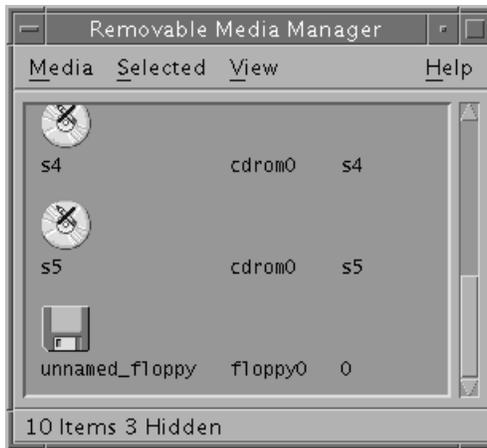


Figure 3.9 Removable Media Manager

Right click on the floppy disk. A popup menu will appear, as shown in Figure 3.10

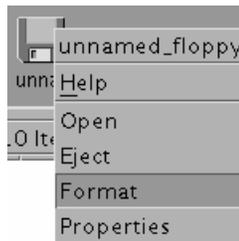


Figure 3.10 Popup Menu for Formatting Floppy Disks

Left click on the Format menu option on this submenu. From here, the Format window opens again and the floppy can be reformatted with any of the available file systems.

Command Line Tools for Floppy Disks

There are some command line tools that can be used to view floppies. If it becomes impossible to look at new floppies inserted into the floppy drive, try these (somewhat cryptic) commands. You can use them at times when the Operating System does not recognize that a floppy has been formatted, added, changed or removed (even with the `volcheck` and `volmgt`) commands.

Lesson 3.10 Working with the Floppy Drive (Command Line)

This lesson describes how to use command line utilities to work with floppy disks. These commands are useful if the Solaris 9 Operating Environment stops working with floppies.

Remember that formatting a floppy disk erases all information on it. Always make sure that there is no critical data on a floppy before you format it.

1. Log in as the root user.
2. Type the command `fdformat -H`
*For problems with formatting a floppy, try the command `fdformat -U -H`
This unmounts the floppy (-U), and formats it in high density (-H).*
3. Type the command `newfs /vol/dev/rdiskette0/<disk-label>` or

newfs /vol/dev/rdiskette0/Unnamed_floppy

The **newfs** command creates a new file system, hence the name "newfs."

4. Type the command **volcheck**

If **volcheck** does not recognize the floppy try these commands:

Type the command **/etc/init.d/volmgt stop**

Type the command **umount -F /floppy/floppy0**

Type the command **rm -rf /floppy**

Type the command **/etc/init.d/volmgt start**

Type the command **volcheck**

5. Type the command **touch /floppy/floppy0/myfloppyfile**

The **touch** command creates a zero-length file with the name that you specify. This will let you list the floppy's contents in the next step.

6. Type the command **ls /floppy/floppy0**

This command shows the contents of the floppy disk.

7. Type the command **cd /floppy/floppy0**

8. Type the command **rm myfloppyfile**

You just removed the file from your floppy disk.

9. Type the command **eject**

The **eject** command tells Solaris 9 that the floppy disk has been removed from the floppy drive.

Using the Find File Utility

The Find File utility is a very nice GUI for finding files. This utility lets the user search for files based on the following criteria:

- **name** The name of the file. This can be the partial name of the file if "Whose name contains" is selected. If the option "does not contain" is selected, the file's name will be excluded from the search. The option "is equal to" lists files and directories with an exact match (includes upper/lower case of the word.)
- **content** The contents of a text file. This search looks through all the specified files for files with the exact text. This search can be case sensitive or not case sensitive.
- **size** Lets you specify if a file should be larger, the same size, or smaller than the size that you enter in the text box.
- **date modified** Search for a file that was modified before or after a certain date.
- **owner** Search for a file based on the owner. The search can also be conducted to exclude files by a particular owner.
- **type** If the file is a directory or not a directory.
- **permissions** - The read/write/execute permissions on the file.

The read/write/execute permissions are based on the owner of the file, or on if all users can read/write/execute the file. Figure 3.11 is a snapshot of the Find Files window.

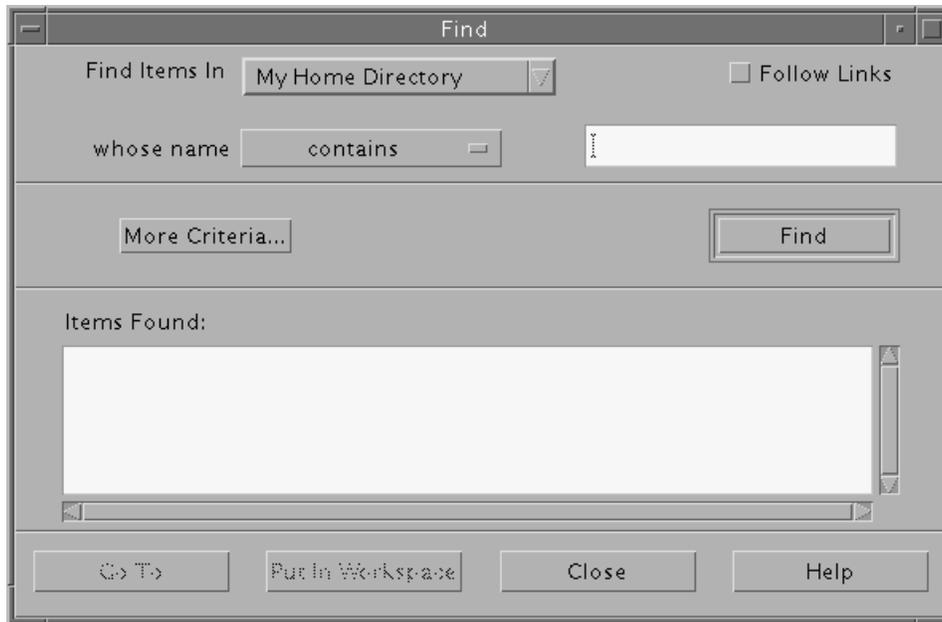


Figure 3. 11 Snapshot of the Find Files Utility

Lesson 3.11 shows how to start the Find Files utility.

Lesson 3.11 Using the Find File Utility

In this lesson readers will use the Find File utility to search for files based on different search criteria. The search criteria include the file's name, the file's content, the size of the file, and the modification date.

1. Log in as the root user.
 2. Right click anywhere in unoccupied desktop space.
 3. Left click on Files.
 4. Left click on Find Files.
- The Find Files window will now appear, as shown in Figure 3.11.*
5. Left click on More Criteria...
 6. Make sure the only criteria selected is "name".
 7. Left click in the text box to the right of "whose name contains" and type in **passwd**
 8. Left click on the drop down box Find Items In.
 9. Left click on the menu choice "Other Folder..."
 10. In the "Selection" text box at the very bottom of the screen, type in **/etc**
 11. Click on the OK button to close the "Other Folder..." popup window.
 12. Left click on the "Find" button.

Several files should be listed, including:

```
/etc/default/passwd
/etc/default/yppasswd
/etc/passwd
/etc/opasswd
```

Because the "whose name" box was set to "contains" any file with the word passwd was listed in the find box, including opasswd and yppasswd. The results can vary, depending on the installation of Solaris 9 NIS and other system software selected.

13. Change the "whose name" drop down box to "is equal to"
14. Press the Find button.
15. In this case only the files with the exact name passwd were selected.

`/etc/default/passwd`

`/etc/passwd`

16. Left click on the drop-down box "Find Items In"
17. Left click on "Other Folder..."
18. Type in `/usr` in the Selection box.
19. Click on the "More Criteria..." button.
20. Make sure that "size" is the only check box button selected.
21. Left click on OK to close the box.
22. Type in 500 as the size. That represents 500 KB in size.
23. A list of files should appear. These files are greater than 500 KB in size.
24. Change the value to 1000 as the size. That represents files that are 1 MB in size.
25. Click on the "More Criteria..." button.
26. Make sure that "name" is the only check box button selected.
27. Left click on OK to close the box.
28. Left click on the "whose name" drop-down box.
29. Left click on "is equal to"
30. Type the name `admintool` in the text box to the right of the "whose name" selection.
31. Left click on the Find button.
When the cursor looks like an hour glass that indicates that the Find program is still searching. Depending on the location of the workstation, the hard drive heads may be heard reading the disk platters very intensely. The cpu/disk activity meter on the Front Panel display may have the disk activity bar running in the red, indicating a high degree of disk activity.
32. Make sure that `/usr/bin/admintool` is highlighted in black.
33. Left click on "Go To"; the `/usr/bin` directory is now displayed in a File Manager window. Left double click on the Admintool icon.
34. Close down the Admintool window, and any other windows that are open ,except the Find window.
A window can be closed by quickly left double clicking on the Menu icon (looks like a dash -) on the top left part of a window.
35. Make sure `/usr/bin/admintool` is highlighted in black
36. Left click on the "Put In Workspace" button.
37. Left double click on the Admintool icon that appears on the desktop.
38. Close all open windows.
39. Select the admintool icon by right clicking on it.
40. Select "remove from desktop" to remove this icon from the desktop.

Using the File Properties Utility

The File Properties Utility is used to view and set permissions on a file or directory. A file or directory can have read/write/execute permissions for a user, a group or all other users on the system. A file can also have what is known as an access control list attached to a file. An access control list gives the system administrator the ability to specify permissions for particular users and groups.

Lesson 3.12 shows how to use the File Properties utility.

Lesson 3.12 Using the File Properties Utility

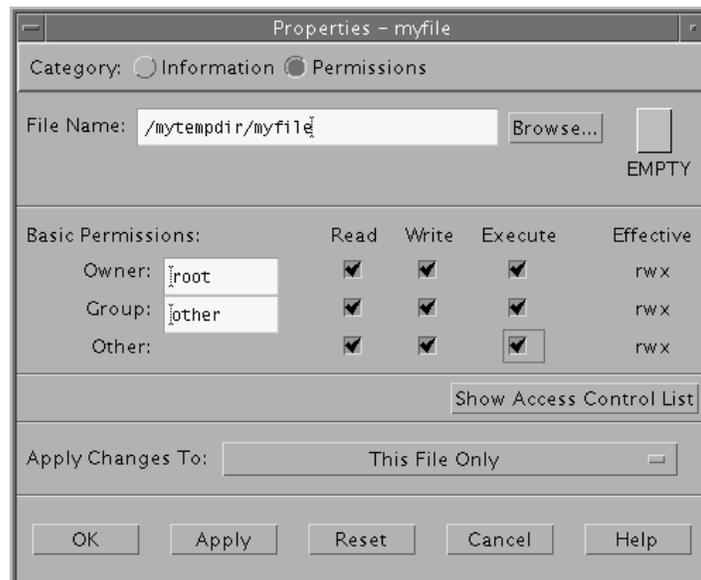
In this lesson readers are going to make a temporary file and directory. The file's permissions are going to be reset with the File Properties Utility.

WARNING

The File Properties Utility is somewhat dangerous in that a system administrator can easily reset permissions on a system folder or critical files and crash a server.

Also, if permissions are set incorrectly, the operating system could be damaged beyond repair. Be extremely careful using this utility!!

1. Log in as the root user.
2. Open a Terminal window.
To open a Terminal window, right click anywhere in unoccupied desktop space. Left click on Tools, then left click on Terminal.
3. Type the command **mkdir /mytempdir**
*The command **mkdir /mytempdir** is used to create a temporary directory named /mytempdir*
4. Type the command **touch /mytempdir/myfile**
*The command **touch /mytempdir/myfile** is used to create a temporary file named myfile in the /mytempdir directory.*
5. Open the File Properties Utility.
To open the File Properties Utility, right click anywhere in unoccupied desktop space, left click on Files, then left click on Properties... A GUI window should appear with the title "Properties."



6. Mark the file permissions Read, Write and Execute for all available permissions.
7. Left click on the OK button
The two previous steps changed the permissions on the /mytempdir/myfile file to rwxrwxrwx.
8. Left double click on the Window Menu buttons on all the displayed windows.
These buttons look like a dash (-) symbol on the top left corner of the window. Double clicking these buttons will kill the window.



Using the Calculator Utility

The Calculator Utility is a scientific/financial/logical GUI calculator. This is very useful for calculating binary to decimal conversions. Figure 3.12 shows the calculator utility.

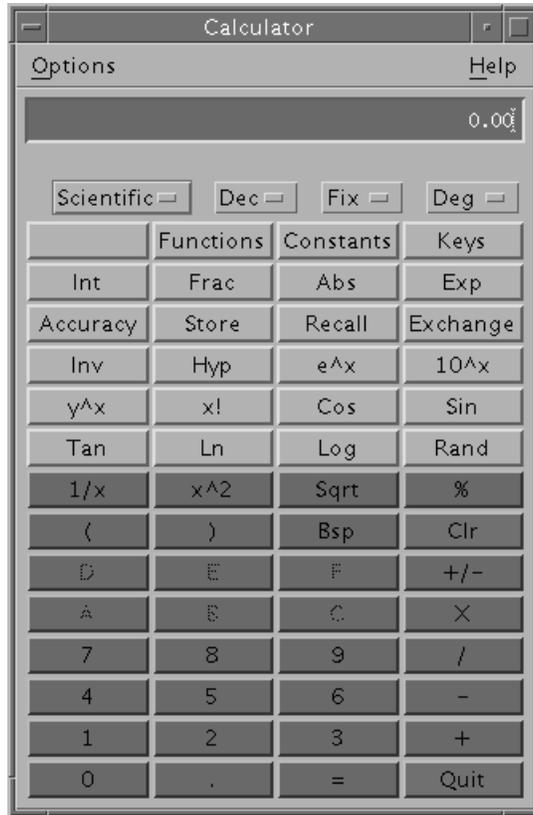


Figure 3. 12 The Calculator Utility

Lesson 3.13 Calculator: Binary to Decimal Conversion

This lesson shows how to convert numbers from binary to decimal and from decimal to binary with the calculator. This skill is absolutely essential when it comes to working with IPv4 and IPv6 networks. In this lesson the number 25 is converted to the Binary number 11001.

1. Logon as user11 or an ordinary user.
2. Right click anywhere in unoccupied desktop space.
3. Left click on Applications.
4. Left click on Calculator.
Figure 3.12 shows the Calculator.
5. Type in the number 25 in the upper text box.
Make sure the Calculator is in Scientific mode.
There are three modes to choose from on the Selection drop down-box: Scientific, Financial and Logical.
6. Left click on the button labeled "Dec -" on the top row.
7. Select "Bin -" from the menu.
The decimal number 25 should now be replaced with the binary number 11001.
A binary number uses the base of 2. Each digit represents a power of two. In this case it is

$2^4 = 16$, $2^3 = 8$ and $2^0 = 1$ so that $16 + 8 + 1 = 25$. Decimal to Binary conversion is discussed in Chapter 27 and is crucial for IPv6 networking.

The Netscape Navigator Browser Version 4.87

Sun Microsystems formed an alliance with AOL (Netscape's parent company) on March 30, 1999. One of the benefits to Sun is that the Netscape Navigator browser version 4.87 is included with Solaris 9 (the browser version could be different with later releases). The Netscape Browser Icon looks like a world globe with red watch hands on top. It is located on the left side of the Front Panel toolbar.



To open Netscape in Solaris 9, left click *once* on the Netscape icon. If the Netscape Icon is clicked on twice, a popup window appears, with a long warning that two copies of Netscape may be running at the same time. When Netscape detects that multiple copies are running under one user, personal certificates (a type of Internet security device) is disabled. This is something you want to avoid.

If your workstation or server is not connected to the Internet, another error message will appear, saying that Netscape is unable to locate the server < www.homepage > and asking you to try again. This message appears when Netscape can not resolve the IP address of the webpage.

To open a webpage that is internal to the workstation or server, click on **File** (on the Menu Bar) then on **Open Page**. From A small popup window appears that looks like Figure 3.13.

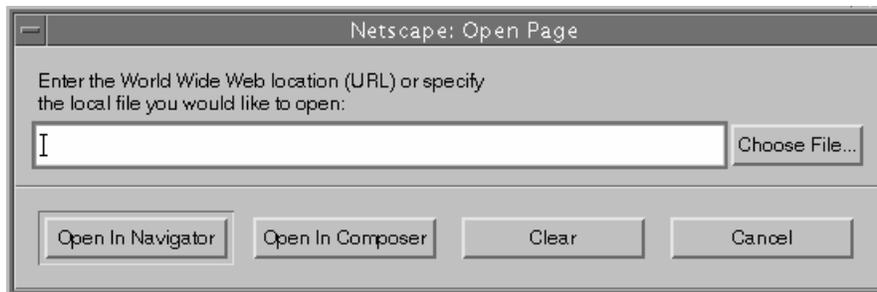


Figure 3. 13 Netscape: Open Page Window

Left click on the **Choose File...** button at the right of the window. To move up a directory, select the double dot (..) option. To move down a directory, double left click on the name of the directory. Any time in Solaris 9 that the double dot (..) notation appears, it means to go up a directory. To open a file, left double click it. The Open Page window will return. It is also possible to type in the path and name of the file directly in the Open Page window, but that become rather fatiguing after a while.

Once a file is selected, left click either the **Open In Navigator** or **Open In Composer** buttons. The **Open In Navigator** button opens the document in the Netscape Navigator. The **Open In Composer** button opens it the Netscape Composer, which lets a user edit the HTML document.

Using the Text Editor

Solaris comes with a command line text editor known as the **vi** editor. This editor does have the advantage of always being there when a GUI display is not available. Unfortunately, the **vi** text editor is a very complex program for new users. There is another text editor supplied with Solaris 9 that is a lot more user friendly. It supports all the usual keyboard edit keys, such as the Backspace, Delete, and Home keys. This text editor is named "Text Editor." It is shown in Figure 3.14.

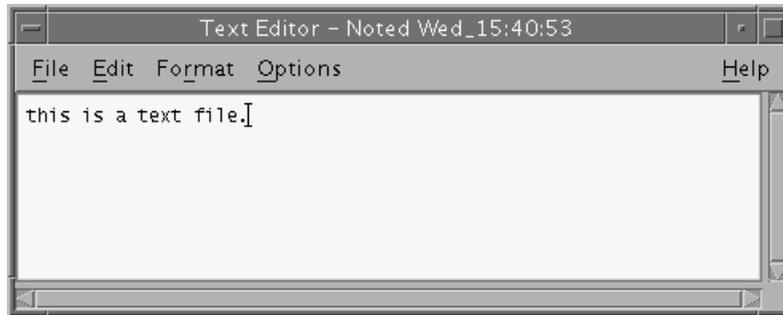


Figure 3. 14 The Text Editor

To open the Text Editor, right click anywhere in unoccupied desktop space. Then left click on Applications. Now, left click on the Text Editor icon.

The Text editor has four main options on the Menu Bar. These options, and some of their suboptions, are:

File - Used to open, save and close files.

- New** - Start a new document
- Open...** - Open an existing document
- Include...** - Insert text from one document into another
- Save** - Save the current document
- Save As...** - Save the current document with a different name
- Print...** - Print the text to the default printer
- Close** - Close the current document

Edit - Used for generic cut and paste options; also supports Find/Change and Check Spelling... Options.

- Undo** - Reverses a typing function
- Cut** - Cut selected text out of the document
- Copy** - Copy selected text
- Paste** - Paste text that has been cut or copied
- Clear** - Clear the selection
- Delete** - Delete selected text
- Select All** - Select all the text in the document
- Find/Change** - Find selected words and/or change the words
- Check Spelling** - Use a spell checker on the document

Format - General page formats such as margins and alignment.

- Settings...** - Adjust the margins and alignment in a paragraph or in the entire document.
- Paragraph** - Apply formatting to a paragraph
- All** - Apply formatting to all the text in the document

.....
Options

- Overstrike** - New characters are inserted over old characters
- Wrap To Fit** - Automatically wraps lines when they approach the end of the window
- Status Line** - Shows the current line number
- Backup On Save** - Each time the file is saved, the previous version is saved with a percent (%) sign appended to the end of the filename. If the

working file becomes corrupted, the older file with the % can be recovered.

Lesson 3.14 Using the Text Editor

In this lesson the reader creates a simple document named “mytestdocument.” This text document only has one line that says “I wrote this with the text editor.” The text document is later opened with the **vi** editor and displayed with the **cat** command.

1. Log on as user11 or a regular user.
2. Open the Text Editor.
To open the Text Editor, right click in unoccupied desktop space. Then left click on Applications and left click on Text Editor.
3. The Text Editor window should have the title "Text Editor (UNTITLED)."
4. Type in the text “I wrote this with the text editor.”
5. Left click on **File**, then Save **As...**
6. Left click in the text box “Enter file name:” and enter the file name “/mytestdocument.”
7. Left click on the OK button.
9. Left double click on the Window Menu buttons on all the displayed windows.
These buttons look like a dash (-) symbol on the top left corner of the window. Double clicking these buttons will kill the window.
8. *To open a Terminal window, right click anywhere in unoccupied desktop space, left click on Tools, then left click on the Terminal icon.*
9. Type the command **cat /mytestdocument**
The text from the mytestdocument should be displayed on the screen.
10. Type the command **vi /mytestdocument**
*The text from the mytestdocument should be displayed on the screen. There should be several tilde symbols (~) under the text. These tilde symbols indicate the lines that don't have text on them. In this case, the **vi** editor has opened the mytestdocument.*
11. Type the command **:q**
*The **vi** command :q quits the document immediately without saving any changes.*



The Style Manager

The Style Manager is used to change the user's desktop settings, such as background image, default font size, keyboard behavior, mouse behavior, the screen saver, the window selection criteria and settings for the Resume/Home session.

Opening the Style Manager is a somewhat indirect operation. To do so:

1. Right click anywhere in unoccupied desktop space.
2. Left click on Tools.
3. Left click on Desktop Controls.
4. Scroll down the window taskbar until an icon with a title of "X Style Manager" (where "X" is another word) is visible. Here, we will use the Color Style Manager icon.
5. Left double click on the Color Style Manager icon.
The Style Manager window will appear. Figure 3.15 shows the Style Manager Window.





Figure 3. 15 The Style Manager Window

The Style Manager icons, from left to right, are:

- Color** Changes the desktop color theme for the windows.
- Font** Changes the default font size and type of font.
- Backdrop** Changes the background image.
- Keyboard** Lets Auto Repeat and keyboard clicks be enabled or disabled (keyboard clicks can be very annoying!)
- Mouse** Adjusts the mouse Double-Click speed, Acceleration, and Sensitivity.
- Beep** Controls the beep that is sounded when an error occurs.
- Screen** Sets the screen saver and also sets a default screen lock time.
- Windows** Determines if a user has to click on a window to make it active or if just passing the mouse pointer over a window will make it active. This icon also determines if a window should be raised or displayed while being moved. Desktop icons can be saved on the desktop or be placed in an Icon Bar. An Icon Bar is a small subwindow that holds the icons.
- Startup** Determines if the CDE should remember the last session (resume current session) or if it should use a default desktop (Return to Home session). This icon also determines if the CDE should ask the user confirm an exit.

Most of the settings on the Style Manager are fairly easy to understand and use. There are some key points about the Style Manager that should be known. The fonts should be adjusted so that the reader can use the CDE comfortably. A small monitor should have large fonts and a large monitor should have small fonts. A different backdrop or background image should be used for every workspace, so that it is easy to tell when the workspace has been changed.

Sometimes co-workers like to play practical jokes on each other. One example is setting the mouse Double-Click speed very high. This could make a company purchase a new mouse, believing that the buttons on the mouse are going bad. This is a very common type of practical joke within the Solaris community, so check this setting. Setting the Beep to a long duration such as 2-½ seconds is also a typical annoyance joke with co-workers. If something goes wrong, the speaker beeps for 2 ½ seconds.

Several screen savers can be selected in the Style Manager. The screen savers work for a short time, then switch to another screen saver. It is a good idea to enable the screen lock. This prevents someone from using your workstation if you accidentally forget to lock it. The Startup icon gives a user the choice of having a default home session or resuming the last session. The Startup icon also lets you define the home session. Most experienced

system administrators have the CDE confirm a Logout because if they accidentally click on the [EXIT] button during an install or other critical admin function, it could cause a great deal of problems.

Lesson 3.15 Using the Style Manager

In this lesson readers open the Style Manager and make changes to their desktop settings.

Please note that when you change a setting in one of the Style Manager windows (such as Screen or Power Management), the change takes place as soon as you click OK in that window. Style Manager never asks for confirmation, and there is no way to "exit without saving changes."

1. Right click anywhere in unoccupied desktop space.
2. Left click on Tools menu choice.
3. Left click on Desktop Controls icon.
4. Scroll down and left double click on the Color Style Manager Icon to bring up the Style Manager window.
5. Double click on Backdrop, select a different backdrop, then click OK when finished.
6. Left click on Screen, click on Screen Saver ON, then click on several screen savers. Click on OK when finished.
7. Left click on Power Management... Select the Disabled setting, click on OK.
8. Enable the Screen Lock for 5 minutes at the bottom of the window, click on OK
9. Left click on Startup and select Resume current session.
10. Select Logout Confirmation Dialog, On.
11. Click on OK.
12. Close all windows that are open by left double clicking on the Window Menu button (looks like a [-] on the top left side of the window).

Understanding the System_Admin Folder Utilities

The System_Admin folder is not really a piece of software or a utility. It is a directory on the Workspace Menu that holds some very nice icons for system administration. To access the System_Admin folder, right click in unoccupied desktop space, left click on Applications, the left click on Application Manager. The Application Manager should have an icon labeled "System_Admin." Left double click on this icon.

The following icons will now appear in a window titled "Application Manager - System_Admin."

- | | |
|--------------------------|---|
| Admintool | Used for administration of users, groups, hosts, printers, serial ports (modem and terminal support) and software packages. This tool is being phased out by Sun Microsystems for the Sun Management Console. |
| AnswerBook2 Admin | This Application only works on systems that have an AnswerBook2 server installed. |
| Disk Usage | Basically shows the output of the df -k command in a terminal window. Pressing a key will close down the Terminal window. |
| Eject | Used to eject floppy disks. It is very confusing and hard to understand, so avoid this utility. |
| Eject CDROM | Rather self explanatory, used to eject CDROMs. |
| Eject Floppy | Properly unmounts a floppy diskette. Understand that even though a floppy diskette can be removed by simply ejecting the disk from the floppy drive, the floppy must be "software ejected." In other |

words, the software has to know the floppy has been removed. The floppy drive does not send an eject notice like the CDROM drive does. When a floppy is removed, a popup window says “/vol/dev/rdiskette0/floppy can now be manually ejected.”

- Font Administrator** Used to install and remove fonts from the CDE desktop. This utility also lets the administrator add a new font path for CDE.
- Format** A very confusing icon for formatting floppy disks. Avoid this cumbersome utility.
- Format Floppy** Brings up the Media Format window. This is the same Media Format window that can be pulled up from the File Manager. Floppy diskettes can be formatted with the PCFS file system, the UDFS file system, the NEC DOS or the UFS (UNIX) file system. Refer to the File System Manager section of this chapter on how to use this utility.
- Open** Lets users view the contents of a floppy drive. Avoid this utility; it is very confusing to work with.
- Open CDROM** Calls on the File Manager to view the contents of a CDROM.
- Open Floppy** Starts the File Manager to view the contents of a floppy diskette.
- Power Management** Opens a dialog box for Power Management. Most companies turn off all forms of power management. Make sure that power management is disabled and don't touch this utility again.
- Print Administrator** Opens the Print Administrator window. The Print Administrator is discussed in detail later in the book.
- Properties** Another poorly created utility. Type the word "floppy" in the text box to see a summary of the floppy drive.
- Protect** Avoid this utility, which can make floppies unstable.
- Removable Media Manager** This is the Removable Media Manager that is used with the File Manager. This is a very good utility to use with CDROMs and floppy diskettes.
- Smart Card** Starts the Smart Card Console. If the system does not have a smart card reader or the server does not use smart cards, don't bother with this utility.
- Solaris Product Registry** A utility for adding and removing software packages from the system. The latest software packages are installed with this utility.
- Suspend System** Saves the system's state and then powers off the system. This utility is discussed earlier in this chapter.

System Load	Shows the load placed on the CPU. It shows a generic display of CPU usage in a small window.
Terminal Console	Opens a Console window. A Console window is different than a Terminal window in that error messages are received in a Console window.
Terminal Remote	Opens a terminal on a remote system. This utility is rather cumbersome and should be avoided.
Terminal Rlogin	Opens a terminal if a foreign host has rlogin enabled. This utility is rather cumbersome, so it should be avoided.
Watch Errors	Displays error messages. It's rather generic and can be avoided.
X server Information	Provides information on the X server that is running the CDE. The X server can be thought of as the graphics engine in the background of the CDE.

Creating a Custom Made Icon

The CDE allows users to create custom-made icons for applications and commands. Although this is usually reserved for software developers, the technique is not all that difficult to master, with a little bit of effort. These icons can then be placed on the desktop of company-wide workstations to open internally produced software. If a company makes a custom database program, this database program can be started by an icon created by the system administrator.

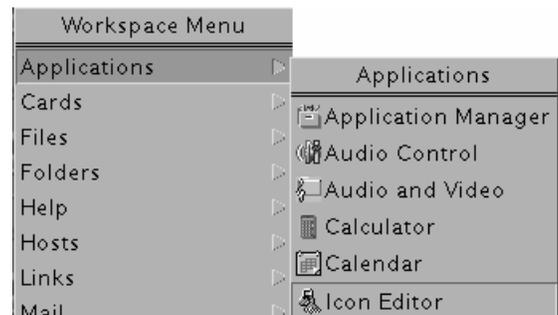
The Icon Editor is a crude bitmap editor. It creates the icon files with the file name **filename.m.pm**. The icons are in the XPM file format. The XBM format can also be used. These icons can be placed in one of the following directories:

/usr/dt/appconfig/icons/C	Any icon placed in this directory is appears with the built-in icons.
/etc/dt/appconfig/icons/C	These are system-wide applications.
\$HOME/.dt/icons	These icons are personal icons.

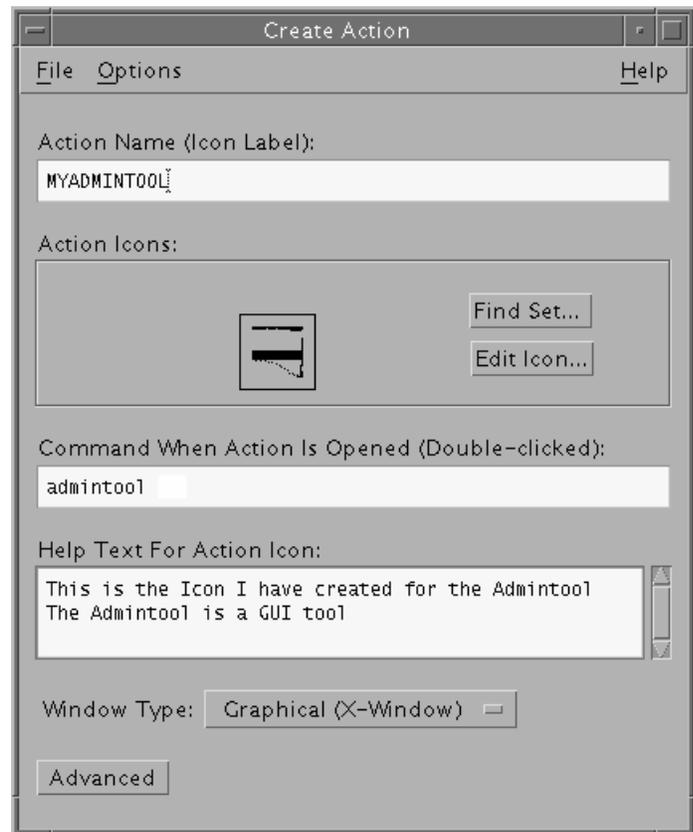
Lesson 3.16 Create a Custom Icon

Where an icon is placed is not very critical, because its location can be specified with the “Create Action” utility.

1. Log on as the root user.
2. Open the Icon Editor.
To open the Icon Editor, right click in unoccupied desktop space, left click on Applications, then left click on the Icon Editor icon.
3. Play with the drawing tools and make any icon desired.
The easiest tool to use is the pencil tool. This tool changes a single pixel's color, depending on what color is selected with the “Static Colors” menu at the bottom of the Icon Editor window.



4. After the icon has been created, left click on **F**ile in the Menu bar.
5. Left click on **S**ave **A**s...
6. Enter the file name as “myicon.m.pm” in the Enter file name: text box. Do not use the double quotes (“) in the name; just type myicon.m.pm as the name.
7. Save the file in the directory
/usr/dt/appconfig/icons/C either by navigating to that directory, or by entering both the path and filename.
8. Left click on the Save button.
9. Left click on **F**ile in the Menu bar.
10. Left click on **E**xit
At this point, in time, the icon myicon.m.pm should be created in the /usr/dt/appconfig/icons/C directory.
11. Open the Workspace Menu by right clicking anywhere in unoccupied desktop space.
12. Left click on the Tools menu item.
13. Left click on the “Create Action” icon.
14. In the “Action name (Icon Label)” box, type in the text MYADMINTOOL.
15. Left click on the “Find Set..” button.
16. Left click on the
/usr/dt/appconfig/icons/C folder in the “Icon Folder” text window.
An hour glass icon should appear for just a moment while the icons are being loaded in the right pane.
17. Scroll down to the bottom of the left pane. The icon created earlier should be there
18. Left click on this icon. Its name should appear in the Enter Icon Filename text window
19. Left click on the OK button to return to the first window.
20. In the text window “Command When Action is Opened” type the command **admintool**
21. In the “Help Text for Action Icon:” window, type the following:
This is the Icon I have created for the admintool. The admintool is a GUI tool used to modify users.
22. The Create Action window will appear.
23. Left click on **F**ile.
24. Left click on **S**ave.
25. A popup window should come up with the message
26. The new action MYADMINTOOL has been placed in your home folder.
27. Left click on OK to close this window.
28. Left click on **F**ile.
29. Left click on **E**xit.
30. Left click on the small triangle above the Home Folder icon in the Front Panel window.
The Folder icon looks like a filing cabinet with some folders sticking out the top.
31. A popup menu will appear above the Folder icon. Left click on the Home Folder version of the File Manager.
32. Scroll up and down the window until the **myicon.m.pm** icon is found.
33. Left double click on the icon.
The Admintool should now start. Use the Admintool, then quit the Admintool.



Administration of the CDE

The CDE is a very complex GUI interface that has a tendency to freeze up. It is important that a system administrator knows how to troubleshoot and repair the CDE. Restoring the server from a backup device is not a good option for repairing CDE problems. The following sections show how to troubleshoot and repair login problems and CDE problems.

Administration of the Login Manager

If for some reason the Login Manager does not appear when the operating system starts up, type the command `/etc/init.d/dtlogin start` to make the Login Manager appear. Type the command `ps -ef | grep dtlogin` to see if the Login Manager's process is running.

If nothing appears on the screen, the Login Manager has not started. Try the command `pkill dtlogin`. If that command does not work, try the command `/usr/dt/bin/dtlogin -daemon; exit` followed by the command `/etc/init.d/dtlogin stop` and then the command `/etc/init.d/dtlogin start` once more.

To make sure the Login Manager starts every time the server reboots, type the command `/usr/dt/bin/dtconfig -e`.

To disable the Login Manager each time the server reboots, type the command `/usr/dt/bin/dtconfig -d`.

Another way to make a server not start the dtlogin process is to rename the `/etc/rc2.d/S99dtlogin` script to another name that does not start with a capital S or K. Solaris 9 ignores any script in the `/etc/rc#.d` directory that does not start with one of these letters. The `S99dtlogin` script starts the Login Manager when Solaris 9 enters run level 2.

The Login Manager is displayed through what is known as the X server. The X server displays the graphics for the Login Manager and the CDE desktop. If the X server is damaged, the Login Manager should give out an error message before dying.

To get information on an error message, go to <http://sunsolve.sun.com> and type in the message. The Sunsolve webpage <http://sunsolve.sun.com> has a free section and a subscription section. The free Sunsolve webpages are rather limited in error messages and resolutions. The subscription section of Sunsolve has some excellent advice. Another possible solution to solving a display problem is to search the Internet through search engines like Yahoo or Excite. Just type in something like `dtlogin` and `error 32` or one or two other words from the error message. Strangely, it's often easier to troll through webpages that you find in a search engine search than to go to Sun's official support webpages.

Administration of the Session Manager

This part of the chapter is not designed to make the reader into a Session Manager Guru. The Session Manager is a very complex set of programs and files. This section only familiarizes the reader with what is going on in a general sense with the Session Manager.

The Session Manager is started from the program `/usr/dt/bin/Xsession` after the Login Manager receives a valid username and password. The Session manager reads the user's `.dtprofile` script in the home directory. The home directory is represented by the variable `$HOME`. So the Session Manager looks for the `$HOME/.dtprofile` file. If it can not find this file, it uses the script `/usr/dt/config/sys.dtprofile`.

After reading the **.dtprofile** or **sys.dtprofile** scripts, the Session Manager reads the following Xsession.d scripts. These scripts set up additional environmental variables and other start daemons. Some of these scripts are read from system wide files for all users, and all CDE sessions and some scripts are customized for a local user. Some of the more common Xsession d files are:

- **.dtpaths** Desktop Search Paths
- **.dtims** Starts the Input Method Server (optional)
- **.dttmpdir** Temporary directory created on a per user, per session basis
- **.xmbind** Reads the \$XMBINDDIR variable and modifies this variable for the user's desktop default.

The Session Manager then reads the Xsession.d scripts located in the **/etc/dt/config/Xsession.d** directory, then the scripts located in the **/usr/dt/config/Xsession.d** directory.

After reading these scripts, the Session Manager starts a welcome message. This welcome message can be customized by changing the **dtstart_hello[0]** variable in the **/etc/dt/config/Xsession.d** file.

The CDE desktop uses what are known as Desktop Search Paths. The CDE desktop trolls through these paths to find desktop applications. Any application that is compatible with the CDE is displayed in the CDE in various menu locations. The Application Manager window usually shows most of the applications found. The paths are used by the Session Manager to search for applications, help files and icons.

The Session Manger then looks at the user's .profile or .login scripts for additional variables and paths. The **.profile** file is created if the user uses the Bourne shell or the Korn shell script. The **.login** script is created for users of the C shell. For this to happen, the DTSOURCEPROFILE variable must be set to true in the **.dtsession** file in the user's home directory. (Yes... this is getting rather complex!)

The Session Manager now runs the ToolTalk daemon. This daemon lets different applications "talk" with each other. After the ToolTalk Daemon starts, the Color Server starts. The Color Server is used to display screen resolutions and colors. Finally, the Workspace Manager is started. The Workspace Manager controls how the windows look and behave.

There are various error logs that can be used to troubleshoot the Login Manager, Session Manager and the CDE desktop. They are

- **/var/dt/Xerrors** shows problems with the Login Manager and Session Manager.
- **\$HOME/.dt/startlog** shows errors reading the **.dtprofile** file.
- **\$HOME/.dt/sessionlogs** shows errors reported from the Session Manager and Window Manager.
- **\$HOME/.dt/errorlog** shows CDE errors while CDE was running.

These log files should give a good clue as to what is happening. Most experienced system administrators just cut and paste the error numbers and error messages onto Internet search engines for webpages and discussion boards that have the same error messages. It's rather rare to have a problem that nobody has seen and resolved before.

Understanding the CDE File structure

The CDE desktop is actually a collection of programs, scripts and files that all work together to present a very nice desktop front. Some common CDE directory locations are:

/etc/dt Holds most of the configuration information for CDE.

/var/dt Used if the CDE desktop needs to create temporary files and directories.

/usr/dt Holds the heart of the CDE system. The **/usr/dt** directory also holds some configuration files for the dt desktop.

\$HOME/.dt Holds user-specific variables and settings.

\$HOME/.dtprofile
A text file that holds each user's specific settings

Key Points to Remember

This chapter demonstrated the different types of applications that are available for the CDE. The CDE desktop is a nice GUI tool to use, but most companies do not use graphical displays on their servers. It is extremely rare to find a server with a keyboard, mouse and monitor attached. The CDE can be displayed on other systems with third party CDE software.

Chapter 4 Working with Files and Directories

Lessons in This Chapter

Lesson 4.1 Basic Directory Commands.....	4-4
Lesson 4.2 Using the touch Command.....	4-4
Lesson 4.3 Using the vi Editor.....	4-5
Lesson 4.4 Vi : Using the h j k l Keys.....	4-6
Lesson 4.5 Vi : Using the .exrc File.....	4-7
Lesson 4.6 Vi : Insert and Append Modes.....	4-7
Lesson 4.7 Vi : The x key.....	4-8
Lesson 4.8 Vi : Opening Files.....	4-9
Lesson 4.9 Vi : Using the \$ ^ and 0 characters.....	4-9
Lesson 4.10 Vi : Using the cw Command.....	4-9
Lesson 4.11 Vi : Using the dw command.....	4-10
Lesson 4.12 Vi : Using the dd command.....	4-10
Lesson 4.13 Vi : Using the J Command.....	4-11
Lesson 4.14 Vi : Using the :r Command.....	4-11
Lesson 4.15 Vi : Searching for text.....	4-12
Lesson 4.16 Vi : Using the G Command.....	4-12
Lesson 4.17 Vi : Scrolling Commands.....	4-12
Lesson 4.18 Vi : Using the ! Command.....	4-13
Lesson 4.19 Vi : Global Replacement.....	4-13
Lesson 4.20 Vi : Advanced File Opening Techniques.....	4-14
Lesson 4.21 Vi : Using + and - Commands.....	4-14
Lesson 4.22 Vi : Marking Text.....	4-14
Lesson 4.23 Vi : Using the C Command.....	4-15
Lesson 4.24 Vi : Using the Tilde (~) Command.....	4-15
Lesson 4.25 Vi : Advanced Delete Techniques.....	4-15
Lesson 4.26 Vi : Using Copy and Paste.....	4-16
Lesson 4.27 Using Text Displaying Commands.....	4-17

Introduction

It is important that a system administrator know how to use command line tools for system administration.. If a server loses its GUI (Graphical User Interface) or does not have a graphics card installed, it is important that a system administrator know how to work with directories and text files. Just ask the simple question, how does a system administrator repair a system that has a problem with the graphical interface like the CDE (Common Desktop Environment), if the GUI interface dies? What if a server never had a GUI interface?

Understand that one of the reasons companies like to use Sun Solaris and other versions of UNIX is because UNIX is designed to be very resistant to hardware failures that would easily crash Microsoft Windows. Part of this “crash resistance” is the fact that Sun Solaris does not need a functional video card or a monitor to work. Most companies do not even connect a monitor to a Sun server. Because of this, a UNIX administrator must be able to work with command line tools. It is extremely unlikely that a company would hire someone to work with Solaris that can not work in a text-only environment.

The first section in this chapter deals with making directories, subdirectories and blank files. In this section, readers will create temporary directories and subdirectories with the **mkdir** command. Next, readers will be shown how to use the **cd** command to move around the directory structure in Solaris. Next, directories will be moved and copied with the **mv** and **cp** commands. Finally, some blank text files will be created with the **touch** command. These files will be copied, moved and deleted within the directory structure.

The second part of this chapter deals with the **vi** text editor. The **vi** text editor is a command line text editor. There is no standard graphical interface to the **vi** text editor. It can be used when there is no GUI display available.

One of the reasons that **vi** is complex is because the standard a-z letters and keyboard symbols are used to give commands to the text editor and add text. The **vi** editor needs to know when a user is writing the characters as text and when characters are being used as a command. For example, the characters **:w** are used as a command to save a file on the hard drive. The only problem is that a user who is writing a program might want to type in an error message that says **Error in :w module.** The **vi** editor needs to know that the **:"w** used here is not a command to save a file.

Because of this dual use of text in the **vi** editor, it has two modes of operation, command line mode and insert mode. This counter-intuitive use of both commands and text makes the **vi** editor very difficult for the novice user to master. This chapter is full of easy-to-follow step-by-step examples to explain the **vi** editor.

The last part of this chapter describes some advanced features of the **vi** editor. These include the ability to change words throughout an entire document and the ability to move around very large documents in a hurry.

Working with Directories

In theory, a hard drive could contain only files. Unfortunately, a hard drive with thousands of files and no directories would become a terrible mess. Sun Solaris 9 uses a rather standard System V Release 4 directory structure.

(Just as a quick side comment, this is pronounced "System Five Release Four" rather than "System Vee Release Four." If you do not want to get pegged as a "newbie" you should always say it this way.)

Directories can only hold files and other subdirectories. The **cd** command is used to move around directories. The **cd** command in Solaris is very similar to the **cd** command in MS-DOS or Microsoft Windows. There are some differences, such as that the **cd** command in Solaris uses a front slash (/) instead of a backslash (\) for the representation of a directory.

For example, in MS-DOS it's possible to have a directory named **c:\mydirectory**. In Solaris, the same directory would be referenced as **/mydirectory**. Also, understand that Solaris 9 does not use drive letters. The directory **c:\mydirectory** does not exist in Solaris 9. There is no such thing as **"c:"** to represent a hard drive. Hard drives are accessed with a "mount point" and not a drive letter. Accessing hard drives and the **mount** command will be covered later in the book. For now, just understand that there are no **c:** or **d:** or **f:** or any other drive letters in Solaris 9.

The **pwd** command is used to show what directory a user is in. This directory is known as the "present working directory" in UNIX. If a user makes a file, it will be created in the present working directory.

The command **ls** shows all the files in a directory. A directory can only contain files and subdirectories. A file can be a text file, a binary file or what is known as a "special file." Special files are primarily used by the Solaris operating system for internal use. Special files will be covered later in some of the more advanced topics chapters.

A subdirectory is a directory under a directory. For example, the `/dog/cat/mouse` directory would have `mouse` as a subdirectory of `cat` and `cat` as a subdirectory of `dog`. If a user type the command `cd /dog/cat` that user would be using an "absolute path name." An absolute path name gives the full directory structure, starting with a front slash (`/`). If a user was in the `dog` directory and typed the command `cd cat` that user would be using a "relative path name." A relative path uses only the subdirectory's name. To get to `/dog/cat` a user must first be in the `/dog` directory before typing the command `cd cat`. Otherwise, how would Solaris 9 know where the `cat` directory is located? Is the `cat` directory under `/etc/cat` or `/dog/cat` or `/usr/cat`? It is the responsibility of the user to specify what directory is being referenced. An operating system can not play a guessing game.

Lesson 4.1 Basic Directory Commands

In this lesson readers will practice moving around the Solaris 9 directory structure with various commands. The `cd` command is used to move around the directories. The `pwd` command is used to show what directory is the present working directory. The `ls` command is used to show the contents of a directory.

1. Log into the system as user11 (this user was created in Chapter 2).
2. When the CDE desktop starts, right mouse click anywhere in unoccupied desktop space.
(Chapter 3 describes the CDE. If necessary, read Chapter 3 before proceeding.)
3. Scroll down the Workspace menu until the cursor is over the Tools menu item.
4. Left click on the Tools menu item.
5. Scroll down the Workspace menu until the cursor is over the Terminal icon.
6. Left click on this icon. This should open up a Terminal window.
7. In the Terminal window, type the command `cd`
When the `cd` command is typed without any arguments (argument = words behind the command) it takes the user to his or her "home directory." A user's home directory can be thought of as that user's "piece of real-estate" on the server. This is where a user has full permission to create, modify and delete files and directories. The root user has (`/`) as the home directory. Most users have `/export/home/<username>` as the home directory. The location of the user's home directory depends on how the server was set up. The user11 account has the home directory under `/usr/user11` as setup in Chapter 2.
8. Type the command `pwd`
The `pwd` command is used to show a user where he or she is in the directory structure.
9. Type the command `cd /etc`
The command `cd /etc` changes the current working directory to the `/etc` directory.
10. Type the command `ls`
This `ls` command shows the contents of the current working directory. In this case the current working directory is `/etc`.
11. Type the command `cd /tmp`
The command `cd /tmp` changes the current working directory to the `/tmp` directory.
12. Type the command `pwd`
The current working directory is `/tmp` as shown by the `pwd` command.
13. Type the command `ls`
The `ls` command without any arguments shows what is in current working directory. The `/tmp` directory is the current working directory.
14. Type the command `ls /tmp`
As can be seen in this example, the command `ls /tmp` shows what is in the `/tmp` directory. In this case the absolute path name `/tmp` is being used.
15. Type the command `cd /usr`
The command `cd /usr` changes the current working directory to the `/usr` directory.
16. Type the command `pwd`
The current working directory is `/usr` as shown by the `pwd` command.
17. Type the command `cd bin`

The command **cd bin** command makes the current working directory **/usr/bin**. In this case the **cd** command is using a relative path name. The absolute path name is **/usr/bin**. If the working directory is **/usr** then the relative path name **bin** can be used to direct Solaris 9 to change to **/usr/bin**.

18. Type the command **cd /etc**
*The command **cd /etc** moved the current working directory to **/etc**.*
19. Type the command **pwd**
*The command **pwd** shows the present working directory. The current working directory is **/etc**.*
20. Type the command **cd default**
*The **/etc** directory has a subdirectory **/etc/default**. The command **cd default** used the relative path name **default** to change the current working directory to **/etc/default**.*
21. Type the command **cd**
*The command **cd** by itself returns the user to the home directory.*
22. Type the command **cd /**
The user is sent to the root directory.
23. Type the command **ls -CF**
*The command **ls -CF** shows a wide listing of all directories and files. The **C** option is used to make the display a wide display. The **F** option puts a **/** character at the end of directories.*

Using the touch Command

The **touch** command can be used to create an empty text file. Imagine opening Microsoft Word and immediately saving a file without any text inside it. An empty text file is the same concept. It is a text file that has no letters and numbers.

The Solaris 9 file system records the last time a file was created, modified or accessed. The **touch** command can also be used to update a file's last modified date.

Lesson 4.2 Using the touch Command

In this lesson the **touch** command will be used to create an empty text file.

1. Log into the system as user11.
2. When the CDE desktop starts, right mouse click anywhere in unoccupied desktop space.
3. Scroll down the Workspace menu until the cursor is over the Tools menu item.
4. Left click on the Tools menu item.
5. Scroll down the Workspace menu until the cursor is over the Terminal icon.
6. Left click on this icon, This should open up a Terminal window.
7. In the Terminal window, type the command **cd**
*When the **cd** command is typed without an argument, it returns the user to his or her home directory.*
8. Type the command **touch myfile**
*The **touch** command creates an empty text file named **myfile**.*
9. Type the command **ls**
*The command **ls** shows what is in the current directory. The **ls** command shows that the **touch** command created this file in user11's home directory.*

Using the vi text editor

Why should anyone learn the **vi** editor? It's very complex, and there are other "nicer" text editors on the market?

That is a question instructors hear all the time. On a strictly professional level, a UNIX system administrator who does not know how to use the **vi** editor would be looked down on by his or her peers. It is a

rather odd “rite of passage” in the UNIX world that a system administrator must know how to use the **vi** text editor. There are other text editors available that are much easier to use but, as unfair as it seems, it is part of the UNIX culture that a system administrator must know how to use it. Also, most companies ask questions about the **vi** text editor during the interview process.

On a practical level, the **vi** text editor is a critical piece of software on a UNIX server. Imagine if a UNIX operating system lost its ability to display a GUI screen and there was not a command line text editor. How would the system administrator change a text file then? Also, terminal sessions and modem connections don’t have a GUI interface available. When it is 3:00 A.M. and a server goes down, a system administrator must be able to connect to the server and change text files. Even if no GUI connection is available, he or she can use the **vi** text editor in this case.

To learn **vi**, the author of this book recommends the book *Learning the Vi Editor* by Linda Lamb, from O’Reilly Publishers. Its ISBN number is 0-937175-67-6, and it has a retail price of \$24.95. This book is excellent! It has a lot of very good material on the **vi** editor. Readers should read Chapters 1 through 4 for basic **vi** questions. Chapters 5 and above deal with advanced topics that are not that critical for readers who are just learning the **vi** editor.

Some drawbacks to this book are that it does not have enough “hand holding” and easy-to-follow lessons in the very beginning. Also, the book covers a generic version of **vi** that does not seem to be 100% compatible with the **vi** editor in Sun Solaris. But once the basics of the **vi** editor are learned, the book can serve as an excellent training/reference material for readers to continue on by themselves.

Try to become familiar with the basics of **vi** before using some of the more advanced features. This editor is extremely counter-intuitive to new users. Also, when it comes to studying advanced features, understand there are some extraneous commands that are not really needed. For example, the command mode key combination “F x” is used to search for the character *x* backward in the current line. Under what circumstances would someone need to use that command? This chapter is not going to describe such oddball commands that a novice user does not need to use at this point in time.

The **vi** text editor is the default text editor installed in Solaris 9. It does not feature any kind of GUI or mouse controls. It can only change text in text files. Most of the file located in the **/etc** directory are text files, such as: **/etc/passwd**, **/etc/groups** and **/etc/shadow** (these three text files define a user). A seasoned Solaris system administrator can edit these files by hand to create a custom-made user, most likely by using the **vi** editor.

The **vi** editor operates in two different modes:

1. Command Mode – **vi** is given commands such as open file, save file, quit. Anytime the **ESC** key is pressed, **vi** goes into command mode.
2. Insert/Append Mode – This is where the user adds text: “My dog likes to chew bones.” When **vi** is in command mode, the user can type any of the letters “**i I a A o O**” to go to insert/append mode.

Lesson 4.3 Using the vi Editor

In this lesson readers learn how to use the **vi** editor. The readers create a simple text file with three lines. The text file is then saved and viewed with the **cat** command.

If for some reason a mistake is made, just type the command **:q** to quit the **vi** editor and then start the lesson again.

1. Log into the system as user11.

2. When the CDE desktop starts, right mouse click anywhere in unoccupied desktop space.
3. Left click on the Tools menu item.
4. Left click on the Terminal icon.
5. Type the command **cd**
6. Type the command **vi textfile**
*This command starts the **vi** editor on a file named **textfile**. By default **vi** starts in command mode. It is expecting the user to perform operations like saving a file, reading a file, moving to a new line, etc. Nothing the user types will be entered as text in the document.*
7. Now type the letter **i**
*The **vi** editor is now in “insert mode.” Now, anything typed will be added to the text.*
8. Type in the following lines of text. Press the Return key after each line.

This is the first line
This is the second line
This is the third line

*If a mistake is made, press the ESC key several times and then type the command **:q** in the **vi** editor. Start the lesson over again from step 6.*

9. If all three line are correct, press the ESC key.
*The **vi** editor is now back in command mode. Remember, when in command mode, **vi** is performing operations (read a file, write a file, quit, etc). Nothing typed now will be saved as text.*
10. Now type **:wq**
*The command **:wq** tells **vi** to write the file to the disk (**w**) and then quit (**q**). A message should appear on the screen that looks like this:*

“textfile” 3 lines, 70 characters

*This is only information about the file created. It gives the name of the file (**textfile**), the number of lines in the file (**3**) and the total number of characters (**70**).*

11. Type the command **cat textfile**
*The **cat** command can be used to view the contents of a text file. In this case the **cat** command is being used to view the text in the file just created. The screen text should look something like:*

This is the first line
This is the second line
This is the third line

12. Type the command **cp textfile textfile.backup**
*The command **cp textfile textfile.backup** makes a copy of **textfile** with the name **textfile.backup**. The file **testfile.backup** will be used in future lessons.*

Lesson 4.4 Vi : Using the h j k l Keys

It is possible to move the cursor with the arrow keys or the **h k l** keys. The reason that the **h j k l** keys are used is that some keyboards and terminals might not properly understand and transmit codes to move the cursor when the arrow keys are pressed. This is very useful if a system administrator comes across a bad combination of keyboard and terminal connection.

1. Type the command **vi textfile**
*Remember that **vi** starts in command mode, so you will not be entering text.*
2. Try any combination of the different keys **h j k l**
h – moves the cursor left
j – moves the cursor down

- k** – moves the cursor up
 - l** – moves the cursor right
3. Type the command **:q**
*The command **:q** tells the **vi** editor to quit **q** immediately and not save any work.*

Lesson 4.5 Vi : Using the .exrc File

The **vi** editor can have options set that control the its behavior. When it starts, it looks for a file named **.exrc** and reads variables from it that affect the way **vi** opens and operates.

1. Type the command **cd**
2. Type the command **vi .exrc**
3. Press the "i" key.
*This puts **vi** in insert mode.*
4. Type the words **set showmode**
This enters the text "set showmode" into the .exrc file
5. Press the ESC key.
*This puts **vi** into command mode.*
6. Type the command **:set all**
*The command **:set all** shows all the possible variables that can be set with the **vi** text editor.*
7. Type the command **:wq**
8. Type the command **vi bigtextfile**
9. Press the letter **i**
*Notice the text "Insert Mode" in the lower right corner. The .exrc file with the line **set showmode** is responsible for the insert messages now seen in the **vi** editor.*
10. Type the command **:q**
*The command **:q** tells the **vi** editor to quit **q** immediately and not save any work.*

Now when the **vi** text editor is used, it will show on the bottom line when it is in insert mode, append mode or command mode.

Lesson 4.6 Vi : Insert and Append Modes

There are six different insert and append commands within the **vi** editor. The insert mode of **vi** starts when the user presses the "i" key, as shown in the first lesson. Table 4.1 shows some other insert/append commands.

Insertion Letter	Type of insertion
I	Insert text after cursor
I	Insert text before cursor
A	Insert text at the start of the line
A	Insert text at the end of the line
O	Open a new line above the current line
O	Open a new line below the current line

Table 4.1 Vi Insert and Append Modes

1. Type the command **vi textfile**
*This command opens the text file named **textfile**.*
2. Move the cursor between the **s** and the **e** in the word "second."
3. Type the letter **O** (small letter **o**, as in **O**scar)
This opens a new line below the second line.

4. Now type the line **This is another line in the text file**

The text file should now look like

This is the first line
This is the second line
This is another line in the text file
This is the third line

5. Press the **ESC** key.
*Any time the **ESC** key is pressed, the **vi** editor goes into command mode.*
6. Now, using the arrow keys, put the cursor just in front of the s character of the word "second" in the file.
7. Now type the command **A**
*This puts **vi** into Append Mode and moves the cursor to the end of the line.*
8. Now type "**of my text file**"
The text should now read:

This is the first line
This is the second line of my text file
This is another line in the text file of my text file
This is the third line

9. Now type the command **:wq**
*This writes the file and then quits the **vi** editor.*
10. Type the command **cat textfile**
*The **cat** command displays the contents of the text file "**textfile**" on the screen.*
11. Type the command **cp textfile.backup textfile**
*The **cp** command in this case is being used to copy the text file **textfile.back** over the **textfile** file. This command restores the original file contents to what they were before the lesson.*
12. Type the command **cat textfile**
*The original contents of the file **textfile** are displayed.*

Lesson 4.7 Vi : The x key

The **x** key is used to delete a single character. It removes the character just before the cursor. If an error is made, the **u** key can be used to "**undo**" it. The **u** command only works immediately after a mistake.

1. Type the command **vi textfile**
*This command opens the text file named **textfile**.*
2. Move the cursor between the s and the e in the word "second."
3. Press the **x** key.
*When the **x** key is pressed, the **e** character disappears. The **x** key is used to delete the next character in front of the cursor.*
4. Press the **u** key
*When the **u** key is pressed, the **e** character reappears. The **u** key is used to undo whatever step was just performed.*
5. Practice using the **x** and **u** keys.
6. Type the command **:q**
*The command **:q** tells the **vi** editor to quit immediately and not save any work .*
7. Type the command **cp textfile.backup textfile**
*The **cp** command in this case is being used to copy the text file **textfile.back** over the **textfile** file. This restores the original file contents to what they were before the lesson.*

Lesson 4.8 Vi : Opening Files

In this lesson readers will learn how to open a file in a directory other than their current home directory. Readers will open a file `/etc/release` with the `vi` editor. The `/etc/release` file is a rather harmless file that gives the name and date of the operating system. Do not modify this file.

1. Type the command `vi /etc/release`
It is possible to open a file in any directory if the full path name is given for that file. In this example the `/etc/release` file was opened. As mentioned above, do not to modify this file.
2. Now type the command `:q`
The command `:q` tells `vi` to not save any changes and quit. Most of the `vi` commands that deal with opening, reading and writing a file require a (`:`). Some `vi` commands require a (`:`) at the start, and others do not. This is something that has to be learned by experience. At this point the rationale behind the colon (`:`) will not be described.

Lesson 4.9 Vi : Using the \$ ^ and 0 characters

Most modern keyboards have the keys **HOME** and **END** to move quickly to the start or finish of a line of text. Instead of using these special keys, the `vi` editor uses the (`$`) symbol to move to the end of a line and the (`^`) symbol or the zero (`0`) key to move to the beginning of a line. These keys work when `vi` is in command mode.

1. Type the command `vi textfile`
2. Try moving the cursor with the `h j k l` keys, and with the `$` and `^` and `0` keys (`0` = number zero).

`h` – moves the cursor left
`j` – moves the cursor down
`k` – moves the cursor up
`l` – moves the cursor right
`$` - moves the cursor to the start of the line
`^` - moves the cursor to the start of the line
`0` – moves the cursor to the end of the line

3. Type the command `:q`
The command `:q` tells the `vi` editor to quit immediately and not save any work .

Lesson 4.10 Vi : Using the cw Command

The `cw` command changes the current word to a different word. The word selected is the word directly under the cursor. The `cw` command finishes the change when the **Return** key is pressed.

1. Type the command `vi textfile`
2. Move the cursor over the `s` in the word “second.”
3. Type the command `cw`
4. Type the word `mouse`
5. Press the **ESC** key.
The text file should look like this

This is the first line
This is the mouse line
This is the third line

6. Type the command `:q`

This command **:q** quits **vi** without saving the text.

7. Type the command **cp textfile.backup textfile**

The **cp** command in this case is being used to copy the text file **textfile.back** over the **textfile** file. This command restores the original file contents to what they were before the lesson.

Lesson 4.11 Vi : Using the dw command

The command **dw** deletes the word directly under the cursor. This is much quicker than pressing the **x** key several times to delete every character of the word letter by letter.

1. Type the command **vi textfile**
2. Move the cursor over the s in the word "second."
3. Type the command **dw**

The text should now look like this

This is the first line
This is the line
This is the third line

4. Type the command **:wq**
This command writes the changes then quits **vi**.
5. Type the command **cp textfile.backup textfile**
The **cp** command in this case is being used to copy the text file **textfile.back** over the **textfile** file. This command restores the original file contents to what they were before the lesson.

Lesson 4. 12 Vi : Using the dd command

To delete an entire line of text from a file, use the command **dd**. The **u** command can undo this if you make an error.

1. Type the command **vi textfile**
2. Move the cursor to the word "second" using the arrow keys or the **h j k l** keys until the cursor is in-between the **s** and the **e**.
3. Type the command **dd**
The text should now look like this

This is the first line
This is the third line

4. Type the command **u** (to undo the change)
The text should now look like this

This is the first line
This is the second line
This is the third line

5. Type the command **:q**
This command **:q** quits **vi** without saving the work.
6. Type the command **cp textfile.backup textfile**
The **cp** command in this case is being used to copy the text file **textfile.back** over the **textfile** file. This command restores the original file contents to what they were before the lesson.

Lesson 4.13 Vi : Using the J Command

The **J** key is used to join two lines together. This is most often used when the Return key is accidentally pressed.

1. Type the command **vi jfile**
2. Press the **i** key
3. Type the following all on one line

This is the first line This is the second line

4. Press the ESC key
5. Place the cursor on the **T** in the second word “**This**” as indicated bellow

This is the first line This is the second line

6. Press the **i** key
7. Press the Return key
The text should look like

**This is the first line
This is the second line**

8. Press the ESC key
9. Move the cursor back to the first character of the first line.
10. Press the **J** key

The text should now look like

This is the first line This is the second line

11. Type the command **:wq**

Lesson 4.14 Vi : Using the :r Command

It is possible in **vi** to read the contents of one file into another file. In this example, the contents of the **/etc/release** file (a text file) will be read into the **textfile** file.

1. Type the command **vi textfile**
2. Move the cursor to the last line and type the letter **A**.
3. Press the Return key twice.
4. Press the ESC key.
The cursor should now be at the start of a blank line, at the end of the text document.
5. Type the command **:r /etc/release**
*This command copies the contents of the **/etc/release** text file into your current file.*
6. Press the ESC key.
7. Type the command **:wq**
8. Type the command **cat textfile**
*The screen should show the original **textfile**, with some extra information about Sun Solaris at the end.*
9. Type the command **cp textfile.backup textfile**
*The **cp** command in this case is being used to copy the text file **textfile.backup** over the **textfile** file.
This command restores the original file contents to what they were before the lesson.*

Lesson 4.15 Vi : Searching for text

The **vi** editor has a built-in word search capability. In this lesson the pattern search command (**/**) is used to find all occurrences of a word. The **n** key can be used to find the next occurrence of a word. The capital **N** key is used to repeat the search in the reverse order.

1. Type the command **man man > bigtextfile**
*This command creates a rather large text file named **bigtextfile**, if it doesn't exist already.*
2. Type the command **vi bigtextfile**
3. Type the command **/displays**
*This tells **vi** to find the word "displays" in the file.*
4. Type the letter **n**
This finds the next occurrence of "displays" in the file.
5. Type the letter **n** several times
This finds more occurrences of the word.
6. Type the letter **N** several time
*Using capital **N** searches backward in the file.*
7. Type the command **?example**
This searches backward and lets you specify the search pattern ("example") in the same command.
8. Type the command **:q**
*The command **:q** is used to quit the vi editor without saving any changes to the file.*

Lesson 4.16 Vi : Using the G Command

With large files, it becomes rather tedious to use the up and down arrow keys to move to a given line. The command **<line#>G** moves the cursor to a line very quickly.

1. Type the command **man man > bigtextfile**
*This command creates a rather large text file named **bigtextfile**, if it doesn't exist already.*
2. Type the command **vi bigtextfile**
3. Type the command **3G**
4. *This moves the cursor to line 3 of the file. The commands shown below move it to lines 10 and 73.*
5. Type the command **10G**
6. Type the command **73G**
7. Type the command **:wq**
*This command writes the file and quits the **vi** editor.*

Lesson 4.17 Vi : Scrolling Commands

This lesson covers the following commands for scrolling through a file:

Command	Cursor Movement
H	To top of screen
L	To bottom of screen
M	To middle of screen
CTRL + F	One screen forward
CTRL + B	One screen backward
CTRL + D	Half a screen forward
CTRL + U	Half a screen backward

Table 4.2 Vi Scrolling Commands

1. Type the command **man man > bigtextfile**
*This command creates the large text file **bigtextfile**, if it doesn't exist already.*
2. Type the command **vi bigtextfile**
3. Type the command **L**
This command moves the cursor to the bottom of the screen .
4. Type the command **H**
This command moves the cursor to the top of the screen.
5. Type the command **M**
This command moves the cursor to the middle of the screen.
6. Now experiment with the commands **CTL+F CTL+D CTL+B** and **CTL+U**.
These commands move the cursor around the screen and scroll the screen up and down.
7. Type the command **:q**
This command quits the vi editor.

Lesson 4.18 Vi : Using the ! Command

It is possible to run a command within the **vi** text editor. This is more of a time saving feature than anything else. Instead of quitting the vi editor, running a command and then starting **vi** again, the exclamation point (!) command breaks out of **vi** and runs a command.

1. Type the command **vi textfile**
2. Now type the command **:!ls /etc**
This command displays the contents of the /etc directory.
3. Type the command **:q**
This command quits the vi editor without saving the work.

Lesson 4.19 Vi : Global Replacement

The **vi** editor has a command option that will search through a line or an entire text file for a particular word or words. After they are found, they can be replaced. For example, it is possible to tell the **vi** editor to replace the word “dog” with the word “cat” on the current line, or all throughout the entire document.

1. Type the command **man man > bigtextfile**
*This command creates the large text file **bigtextfile**, if it doesn't exist already.*
2. Type the command **vi bigtextfile**
3. Now type the command **/markup**
This command finds the next occurrence of "markup"
4. Now type the command **:s/markup/alter**
*The command **:s/markup/alter** changes the **first** occurrence of "markup" to "alter" in the current line. If "markup" occurred **more than once**, only the **first** occurrence would be changed.*
5. Now type the command **1G**
*The command **1G** takes the cursor to line 1*
6. Now type the command **:s/e/z/g**
*The command **:s/e/z/g** changes **all** occurrences of the letter **e** to **z** on the current line (because of the /g). This does not change all occurrences of the letter **e** in the document, just on the current line.*
7. Now type the command **1G**
*The command **1G** takes the cursor to line 1.*
8. Now type the command **:%s/e/z/g**
*This command changes the letter **e** to **z** throughout **the entire document**. The /g makes it change all occurrences on a line, and the % makes it search the entire document, not just the current line.*
9. Now type the command **:q**
This command quits the vi editor and does not write the changes to the file.

Lesson 4.20 Vi : Advanced File Opening Techniques

There are several convenient ways to open the **vi** text editor other than the default edit mode. Rather than opening the file to the first line, it is possible to type the following commands:

vi +<line-number> <filename> starts on a specific line number.
vi /<search pattern> <filename> opens the file and finds the search pattern.

1. Type the command **man man > bigtextfile**
2. Type the command **vi +10 bigtextfile**
*This command opens **bigtextfile** with the cursor on the tenth line.*
3. Type the command **:q**
4. Type the command **vi /displays bigtextfile**
*The command **vi /displays bigtextfile** opens **bigtextfile** and moves to the first occurrence of "displays."*
5. Type the command **:q**

Advanced Vi Editing

This part of the chapter deals with some advanced features of the **vi** editor. These features can be thought of as "speed techniques." These commands let you delete several characters or lines of text in just a few keystrokes. Understand that the previous material is more than sufficient for day to day UNIX administration. These techniques are only convenient commands for advanced **vi** users.

Lesson 4.21 Vi : Using + and - Commands

In this lesson the readers learn that the + and - keys are very convenient for quickly moving to the first character in the next or previous lines.

1. Type the command **man man > bigtextfile**
2. Type the command **vi bigtextfile**
3. Practice using the + and - keys
Always use the keys at the top of the keyboard for this. The + and - keys on the number pad do not work in this example.
4. Type the command **:q**

Lesson 4.22 Vi : Marking Text

In this example the user will experiment with marking text and returning to the marked text. The advantage to marking text is that the cursor can be quickly returned to the previous spot. This is a nice feature when it comes to reading through a document. The disadvantage of marking text is that it is so easy to forget where all the marks are.

1. Type the command **man man > bigtextfile**
2. Type the command **vi bigtextfile**
3. Type the command **13G**
*The command **13G** moves the cursor to the 13th line.*
4. Type the key combination **ma**
This creates an invisible mark called "a" at the start of the 13th line.
5. Type the key combination **20G**
*The command **20G** moves the cursor to the 20th line.*
6. Type the key combination **mb**

This creates an invisible mark called "b" at the start of the 20th line.

7. Type the key combination `'a` (an apostrophe or "single quote" followed by an `a`).
The command `'a` moves the cursor back to the "a" mark that was created with the `ma` command.
8. Move to any line in the document, then press the `'b` key combination.
The command `'b` moves the cursor back to the "b" mark that was created with the `mb` command.
9. Type the command `:q`

Lesson 4.23 Vi : Using the C Command

The `C` key can be used to change the rest of a line.

1. Type the command `man man > bigtextfile`
2. Type the command `vi bigtextfile`
3. Type the command `30G`
The command `30G` moves the cursor to the 30th line.
4. Move four characters to the right
To move four characters to the left, press the `L` key several times, or press the right arrow key several times.
5. Press the `C` key
There should be a dollar sign (\$) sign at the end of the line. This means that everything between the cursor and the dollar sign (\$) will be changed to any new text that is typed in.
6. Type `with some more text`
The remainder of the line is replaced with the text that you typed.
7. Type the command `:q`

Lesson 4. 24 Vi : Using the Tilde (~) Command

The tilde key (`~`) can be used to change the upper/lower case of a letter.

1. Type the command `man man > bigtextfile`
2. Type the command `vi bigtextfile`
3. Move to any letter in the `bigtextfile` file.
4. Press the tilde key (`~`) several times.
The tilde key toggles a letter between the upper case and the lower case.
5. Type the command `:q`

Lesson 4.25 Vi : Advanced Delete Techniques

In this lesson the user will try using some advanced delete commands within the `vi` editor. The delete commands can be used to delete several characters, words, or the end of a sentence, with just a few keystrokes.

1. Type the command `man man > bigtextfile`
2. Type the command `vi bigtextfile`
3. Type the command `30G`
The command `30G` moves the cursor to the 30th line.
4. Type the command `4x`
The command `4x` is used to delete the next 4 letters. Whenever a number is in front of a command, the command is repeated that number of times. In this case, `4x` tells the `vi` editor to use the `x` (delete character) command four times.
5. Press the `u` key
The `u` command tells the `vi` editor to undo the last command. In this case, the `4x` is undone.
6. Press the `D` key.

7. Notice that this is capital D. The **D** command deletes the entire line that the cursor is in.
8. Press the **u** key
This undoes the last command (undeletes the line).
9. Move the cursor to different lines and practice using the capital **D** and **u** commands.
10. Type the command **d\$**
*The command **d\$** instructs the delete function to delete to the end of the line symbol (\$).*
11. Press the **x** key.
*Notice that this is capital X. The **x** key is different than the lower case **x** key in that it deletes the character before the cursor.*
12. Type the command **:q**

Lesson 4.26 Vi : Using Copy and Paste

In this lesson readers will learn how to copy and paste text in the **vi** editor using the **Y** and **P** keys.

1. Type the command **vi textfile**
2. Create the text lines (if they don't exist)

This is the first line
This is the second line
This is the third line

3. Move the cursor to the first character of the second line.
4. Press the capital **Y** key.
This copies the entire line.
5. Move the cursor to the first character in the third line.
6. Press the capital **P** key.
*This pastes the copied text into the file. At this point, the file **textfile** should look like:*

This is the first line
This is the second line
This is the second line
This is the third line

The second line should now be pasted under the third line.

7. Type the command **:q**

Examining Text Files

There are several tools that are used to view text files. The **head** command displays the first few lines of a text file. If no options are specified the **head** command shows the top ten lines of a file. A user can specify the number of lines with the options **head -# <filename>**. To view the top ten lines of all files in a directory, type the command **head ***.

The **tail** command displays the last lines of a text file. If no options are specified the **tail** command shows the bottom ten lines of a file. A user can specify the number of lines with the options **head -# <filename>**. To view the bottom ten lines of all files in a directory type the command **tail ***.

The **more** command is used to display text one screen full at a time. The **cat** command is used to display small text files.

The **sed** command lets you look at a range of lines in a file.

Lesson 4.27 Using Text Displaying Commands

This lesson shows reader how to use the **head**, **tail**, **sed**, **more** and **cat** commands to look at text files on the system.

1. To view the top 10 lines of a text file called `/etc/passwd`, type the command **head /etc/passwd**
2. To view the top 5 lines of `/etc/passwd`, type the command **head -5 /etc/passwd**
3. To view the bottom 5 lines of `/etc/passwd`, type the command **tail -5 /etc/passwd**
4. To view lines 10-15 of `/etc/passwd`, type the command **sed -n '10,15p' /etc/passwd**
5. Type the command **man man > bigtextfile**
6. Type the command **more bigtextfile**
*The **more** command lets you view a large file without using an editing program.*
7. Press the **Return** key several times. This moves you down one screen at a time.
8. Press the **b** key several times to scroll back up.
9. Press the **space** bar several times to scroll down one line at a time.
10. Press the **q** key to quit the **more** command.
11. To view a small file, (such as `/etc/passwd`) type the command **cat /etc/passwd**
12. To open a file and move directly to the first occurrence of a search pattern, start **vi** with an argument such as **vi +/Invocation bigtext**
*This opens the **bigtext** file and moves to the first occurrence of "Invocation" in the file.*

Key Points to Remember

This chapter demonstrated how to work with files and directories. It is important that the user spend a great deal of time working with the **cd**, **pwd**, **ls** and **vi** commands. The UNIX file system is very different from the Microsoft Windows file system, so if you are used to Windows, you should make enough practice time to learn the Solaris 9 commands.

Chapter 5 Getting Help

Lessons in This Chapter

Lesson 5.1 Working with the man Command	5
Lesson 5.2 Saving Man Pages to a Text File	5-6
Lesson 5.3 Using the grep Command With the man -k Command	5-7

Introduction

There are various sources for help with Solaris 9. Just as a quick reference, the primary sources for help are:

Web Based

There are three webpages that deal with help issues. Understand that these webpages tend to overlap with the same information. Some information and reports can be found on all three webpages or on just one. The reader needs to visit these webpages and gain firsthand experience. Sun Microsystems updates these webpages constantly, so don't be too surprised if the information changes from month to month.

<http://www.sun.com/bigadmin>

This webpage has a lot of FAQs, example scripts, patches, patch reports, etc.

<http://docs.sun.com>

Provides online manuals in the form of HTML webpages or Adobe Acrobat files.

<http://sunsolve.sun.com>

This service is broken down into free webpages and pay-for-service webpages that require a contract and an annual fee.

CD-ROM Based:

Both the Ship Kit and the System Administrator's Media kits have two documentation CD-ROMs. These CD-ROMs tend to become quickly out of date due to the fact that Solaris changes rather quickly. Most companies have Internet connections, so the CD-ROMs should only be used on systems that are isolated from the Internet.

Documentation on CD-ROM

Two CDRoms are provided with both types of Solaris 9 Media Kits. These CDRoms have both HTML and Adobe Acrobat .pdf files.

System Based

Man pages come with Solaris 9 and with third-party software. They are very nice because they can be viewed on systems that don't have graphical displays. The other nice benefit to man pages is that they can be read very quickly. Just

type the command **man <keyword>** and in one second the documentation is on the screen. HTML help files and Adobe Acrobat **.pdf** files are also a good help source, but it becomes a real pain to have to wait for the Netscape browser or Acrobat reader to start up each time. Once the document is displayed, the system administrator needs to troll through the document for a particular command and its options.

man pages

Used when a system does not have a graphical display. The **man** command is used to display man pages on a terminal or console.

Each of these sources for help has its place. The webpages from Sun are good for troubleshooting and picking up advice. The *on-system* documentation is necessary for systems that are not connected to the Internet. If the network DNS server goes down (required for Internet access), how would a system administrator look up Solaris documentation online? It is important to have local documentation on the server. This will save time when it comes to troubleshooting. The final source of help, man pages, are very convenient when a system administrator needs to look up a particular command. Imagine trying to sift through a three hundred page document just to find the command **ls -A**.

Online Documentation

This section describes the three best webpages for online documentation. Try to use official Sun webpages instead of other third-party webpages. The Sun webpages are technically accurate, while third-party webpages on Sun topics tend to be hit or miss when it comes to accuracy and quality.

The <http://docs.sun.com> Webpage

This is the primary source for online manuals and official Sun documentation. This documentation does not include FAQ information. This documentation comes in the form of HTML webpages and Adobe Acrobat files. The best practice is to download the Adobe Acrobat files associated with a particular topic.

Quick Tip

- To download Adobe Acrobat files, navigate to a desired guide or chapter. Left click on the tab labeled “Download PDF” located on the top right of the webpage.



- Downloaded files may have odd names like **815-2318.pdf**. Rename them to something normal like **Solaris9-DNS-server.pdf** so that you can find them easily later on. You can use any filename, but make sure the file extension is **.pdf** so that the Adobe Acrobat Reader will recognize the file as an Acrobat file.
- The Adobe Acrobat Reader does not come with Solaris 9. You can download it from <http://www.adobe.com>. As of the publication date of this book, there is no version of the Adobe Acrobat Reader for Solaris 9, but the Adobe Acrobat Reader for Solaris 8 works just fine.
- Chapter 3 describes how to download and install the Adobe Acrobat Reader.

The <http://sunsolve.sun.com> Webpage

This webpage has a free section and a pay-for-service webpage. The free webpage has Solaris 9 patches, free support documentation and security information alerts. The pay-for-service section requires what is known as a “Sunsolve contract.” Usernames and passwords are given out only to employees of companies that have a Sunsolve contract. Some key points:

download patch cluster

Left click on "patches" (located on the left side of the webpage)
Under the title “Solaris Patches” left click on
“Recommended and Security Patches.”

Searchable Collections

Left click on this hyperlink to view the free collection of Info Docs, Patch Reports, iPlanet Articles, and Symptoms and Resolutions. The documents on the free webpage are not as good as the pay-for-service documents.

The <http://www.sun.com/bigadmin> Webpage

This is the best webpage for new system administrators. The webpage is designed to give tips and hints on how to use Solaris 9. This webpage features discussion boards, FAQs, Scripts, etc. This webpage specializes in the latest information on Solaris and other Sun products.

Using Man Pages

Man pages are help text documents that can be viewed with the **man** command. They give useful information for the various commands that are available in Solaris 9. They are needed if a system does not have a GUI.

Almost every command and important file has a matching man page. Most of the files in the **/etc** directory have matching man pages. For example, the file **/etc/resolv.conf** is a text file located in the **/etc** directory. There is no **resolv.conf** command, but there is a man page associated with the **resolv.conf** file. When companies produce software packages, they typically produce man pages that relate to their packages.

The command **man -k <keyword>** can be used to search through all the man pages with a keyword search. If for some reason the system does not allow the **man -k <keyword>** search, type the command **catman -w &** before using the **man -k** command. The **catman** command creates the **windex** database file. The **windex** database lists all the man pages found on the system and is critical for the **man -k** command. The command **apropos** performs the same function as the **man -k** command.

The man pages are located in the directory **/usr/share/man**. There are two different types of raw source files for the man pages: nroff and SGML (Standard Generalized Markup Language) source files. The nroff source files are located in the **/usr/share/man/man*** directories. The SGML source files are located in the **/usr/share/man/sman*** directories. For example, the directories **/usr/share/man/man1**, **/usr/share/man/man2** and **/usr/share/man/man3** are located in the **/usr** directory. These directories are the source directories for the first, second and third sections of the nroff source files.

The Environmental Variable **MANPATH** is used to define other directories that the **man** command should search through when it is trying to find a man page. The file **/usr/share/man/man.cf** is used to define the sections of the man pages. This file has the format

```
MANSECTS=section,section,section, etc...
```

Man pages are divided into sections. These are:

NAME

The name of the command and very similar commands This includes a very short text description of the command.

SYNOPSIS

Gives the absolute path name for a command, for example **/usr/bin/ls**. The SYNOPSIS also shows all the options that are available with a command, such as: **-s, -r, file**, etc.

DESCRIPTION

Several paragraphs are printed about the command or file, related files and other general information.

OPTIONS

Shows the different command line options that are available with each command. For example, the **ls** command supports the **-A** option. The option is followed by a brief paragraph describing what that option does.

OPERANDS

Describes such things as filenames, keywords and reserved words that follow the options.

USAGE

A description of what the command or file is used for. This usually includes some other related files and commands.

EXAMPLES

Shows short example lines, with some text explaining what the example would accomplish.

ENVIRONMENTAL VARIABLES

The environmental variables that affect the command. For example the **MANPATH** variable affects how the **man** command looks for man pages.

EXIT STATUS

When a command does not properly exit, it sometimes sets what is known as an “exit status” or number. This number represents a type of error. This section shows the exit numbers and the meaning of the exit status. In most cases zero (0) indicates success and any exit status greater than zero indicates a failure.

FILES Other files and commands that are similar to the command being referenced in the man page.

These sections are not hard coded in stone anywhere. Other section titles like NOTES or SEE ALSO, are sometimes used.

The following list shows some options that can be used with the **man** command.

man <options> command

<options>

- a** Displays all man pages that match the command. Sometimes there is more than one man page for a command. This displays all the man pages found for a command.
- f** Prints a one-line summary of the command.
- F** Forces the **man** command to look through all the directories in the **MANPATH** environmental variable and the **man.cf** file.
- k** Searches through all the man pages for a keyword.
- M </search/path/for/man/pages >**
A command line search path for man pages.
- s** Specifies a section of the **man** page to search through.

Lesson 5.1 Working with the man Command

In this lesson readers are going to work with the different options that are available with the **man** command. These options give the **man** command additional flexibility in searching for keywords or showing additional man pages.

1. Log in as the root user
2. Open a Terminal window

To open a Terminal window, right mouse click anywhere in unoccupied desktop space. Left click on Tools, then left click on the Terminal icon.

3. Type the command **man ls**
*The command **man ls** opens the man page associated with the **ls** command.*
4. Press the **space bar** several times.
*Each time the **space bar** is pressed, the man page scrolls down one page.*
5. Press the **b** key
*Each time the **b** key is pressed, the man page scrolls up one page.*
6. Press the Return key
Each time the Return key is pressed, the man page scrolls down one line. When the end of the man page is reached, the terminal prompt reappears.
7. Type the command **man ls**
8. Press the **q** key
*The **q** key exits the man page immediately, without having to scroll to the end.*
9. Type the command **man -k floppy**
10. *The command **man -k** searches through all the man pages for any command or file that has anything to do with a floppy. The command will most likely fail with an error message. We will see below how to make this work.*
11. Type the command **catman -w**
*The command **catman -w** creates the windex database. This is necessary if the **man -k** search will be used in the future. Type this command if extra man pages have been added to the system.*
12. Type the command **man -k floppy**
*Now all the files associated with a floppy can be seen. Anytime a new server is created, don't forget to type the command **catman -w** to create the windex database file that is needed to search through man pages.*
13. Type the command **apropos floppy**
*The command **apropos** does the same thing as the **man -k** command.*
14. Type the command **man -s5 largefile**
*The **-s5** option tells the **man** command to look through the fifth section of the man page for **largefile**.*
15. Type the command **man -a lp**
*This displays all the man pages associated with the **lp** command.
Press the **space bar** several times. Notice that two different **lp** man pages are being displayed. If for some reason the man page for a command does not seem to relate to the command, try the **man -a** option to see if any other man pages are associated with the command.*
16. Type the command **man ls lp mv**
*In the above example, several man pages are displayed one after the other: the **ls** man page, the **lp** man page and the **mv** man page.*
17. Type the command **man -f lp**
*The **-f** option displays only a short description of the command. It shows the NAME section of the man page associated with the command.*

Lesson 5.2 Saving Man Pages to a Text File

This lesson shows readers how to save the text in a man page to a text file. This is convenient when the contents of the text file needs to be cut and pasted into another text file. This is extremely valuable if the system administrator needs to create some documentation for junior administrators or workstation users.

1. Log in as user11 (this user was created in Chapter 2).
2. Type the command **man passwd**
3. Type the command **man passwd > mypasswdtextfile**
*The command **man passwd > mypasswdtextfile** sends the standard output of the man passwd command into a text file named **mypasswdtextfile**.*
4. Type the command **vi mypasswdtextfile**
*The **vi** text editor is being called on to read the **mypasswdtextfile***

5. Type the command **q!**
6. Type the command **man ls | col -b > mylstextfile**
*The command **man ls** is piped | into the **col** command. The **col -b** command cleans out the rather unpleasant looking **^H** symbols in the text file. The pipe symbol (|) is created by holding down the **SHIFT** key and the backslash (\) key.*
7. Type the command **more mylstextfile**
*This shows **mylstextfile**, one screen at a time.*

Lesson 5.3 Using the grep Command With the man -k Command

Occasionally the **man -k** command produces so much output that everything scrolls by before it can be read. The **grep** command is used to extract only output that the reader wants to see.

1. Log in as user11 or a regular user (user11 was created in Chapter 2).
2. Type the command **man -k df**
*The **df** command is used to show the free space on a disk. In this example the output from the command **man -k df** goes by too quickly.*
3. Type the command **man -k df | grep disk**
To break this command down into its elements:
man -k df - Search all man pages for the **df** command
 | - The pipe symbol is used to pipe the output of the **man -k df** command into the **grep** command.
grep disk - The **grep** command takes all the output from the **man -k df** command and pulls out only the lines that have the word "disk" in them.

Adding Man Pages to a System

There are times that a software package is installed on a server and the man pages are just copied to a directory name, such as **/opt/<software_directory>/man**. The man pages can not be seen or used.

To fix this, log on as the root user. Type the command **cd /<software_directory>/man**. Under this directory there are usually subdirectories named **man1m**, **man2m**, **man3m**, etc. The paths to these directories are

```
<software_directory>/man/man1m
<software_directory>/man/man2m
<software_directory>/man/man3m.
```

and so forth.

Just copy the man pages in these directories to the corresponding directories in the **/usr/share/man/man*** directories. Run the command **catman -w** again to include these man pages in the windex database.

If for some reason these tricks don't work to natively incorporate the man pages into the directory, modify the **MANPATH** variable for the user. Put the line

```
MANPATH=$MANPATH:/software_directory/man ; export MANPATH
```

in the **.profile** file for the user. This will let the user see the man pages. The **MANPATH** line can also be added to the **/etc/profile** file so that everyone will see the man files.

Key Points to Remember

This chapter focused on some of Sun's most popular webpages. Sun has a tendency to make a lot of changes to their webpages. There could be different types of webpages in the future that not covered by this text. Remember, the <http://sunsolve.sun.com> webpage has a free section and a pay-for-service section that require a service contract to enter. This chapter also covered the **man** command in detail. Whenever you run across a Solaris command or configuration file that you need more information on, use the **man** command to access that information. If that doesn't work, use the **man -k** option to find related man pages.

Chapter 6 Solaris Bootup and Shutdown

Lessons in This Chapter

Lesson 6.1 Setting OpenBoot Variables
Lesson 6.2 How to Work with Run Control Scripts 6-8

Introduction

This chapter covers the steps that the boot process takes when Solaris 9 starts. One of the key differences between Solaris 9 and Microsoft Windows and Novell NetWare is that Sun Microsystems gives the system administrator tremendous access to the software code that produced the Solaris 9 operating system. A system administrator can customize a server into an incredibly large number of configurations. A high level system administrator can create custom scripts and programs that interact directly with operating system. This almost makes the server into a completely new operating system based on Solaris 9

This chapter covers the boot process and the shutdown process. The boot process lasts from the time the system is first booted until the system comes to its default run level (usually the third run level). After describing the boot process, it will cover how readers can customize their own workstations.

The Four Phases of the Boot Process

1. Boot PROM phase The hardware tests itself and initializes itself.
2. Boot programs phase The initial boot programs are loaded into memory.
3. Kernel phase The Kernel loads itself and its modules into memory and then unloads the boot programs from memory.
4. Init phase The `init` process is started by the Kernel. The `init` process then executes the run control scripts.

Phase 1 The Boot PROM Phase

During this phase of the boot-up, the system first powers up and checks itself. On the PROM chip is a program known as the monitor program. This program is used for the initial system tests and diagnostics. It tests the system's memory, CPU and motherboard(s). It does not test all devices attached to the server, only the server's main components.

If a third-party device is attached to an SBus controller, the device driver is then loaded from a firmware chip on the device (some manufacturers don't include device drivers on the hardware itself.) If the OpenBoot variable `diag-level` is set to `max` and the variable `diag-switch` is set to `true` the system will perform extensive diagnostics during the power on self test.

During the initial hardware tests, the information from the `banner` command is displayed. The banner information looks like Figure 6.1

**Sun Blade 100 (UltraSPARC-IIe), Keyboard Present
OpenBoot 4.0, 128 MB memory installed, Serial # 50632835.
Ethernet Address 0:3:ba:2:c2:3d, Host ID: 8323c12b.**

Figure 6.1 Output from the banner Command

After the power on self test is complete, the boot process stops at the **OK** prompt or continues to boot the Solaris 9 operating system. This depends on the value of the OpenBoot **auto-boot?** variable:

- If the **auto-boot** variable is set to **true** the system boots the device specified in the **boot-device** variable. The default boot device OpenBoot value on most system is the **disk** or **disk:a**. A second typical boot device (**net**) can also be specified. If for some reason the first boot device does not work, the second boot device is tried.
- If the **auto-boot?** variable is set to **false** the system stops at the **OK** prompt.

Lesson 6. 1 Setting OpenBoot Variables

In this lesson, the reader will practice working with various OpenBoot variables. This will change the boot behavior of a SPARC system. Please note that Intel based systems do not have the **OK** prompt or OpenBoot variables.

1. Power on the SPARC system. If it starts loading the operating system, halt this process. Press the **STOP + A** keys simultaneously on a Sun keyboard, or press the **CTL + Break** keys simultaneously on a PC keyboard.
*At this point, only the **OK** prompt should be visible on the screen.*
2. Type the command **printenv**
*The command **printenv** shows all the variables from the **OK** prompt. Press the **Return** key to scroll down one line at a time. Press the **spacebar** to scroll down several lines at a time.*
3. Type the command **setenv auto-boot? false**
*The variable **auto-boot?** is used to determine if the system should boot the operating system automatically. If the **auto-boot?** variable is set to false the SPARC system will halt at the **OK** prompt.*
4. Type the command **printenv boot-device**
*When the **printenv** command is used with a variable, only that variable's value is printed.*
5. Type the command **devalias disk**
*The **devalias** command shows what the device alias **disk** actually means. A device alias is a human-readable name that refers to a device tree. The device alias **disk** is the physical device path to the hard drive. This was set by the Solaris 9 operating system when Solaris 9 was being installed.*
6. Type the command **reset-all**
*The command **reset-all** simulates power cycling the system.*
7. Insert the Solaris 9 Install CD-ROM in the CD-ROM drive.
8. Type the command **boot cdrom -s**
*The command **boot cdrom -s** is used to boot a stripped-down version of the Solaris 9 operating system from the CDROM. From here it is possible to perform maintenance on the server's inactive disks. There should be a message **INIT: SINGLE USER MODE***
9. Type the command **format**
The Available Disk Selections menu appears. Choose any of the available disks by typing its number and pressing Return. The Format menu appears. It includes choices for setting and displaying disk information, formatting or repairing a disk, and other operations. Don't choose any of these choices!
10. To exit the Format menu, type the command **quit**
11. Type the command **init 0**
This command gracefully shuts down the operating system.

12. Type the command **setenv diag-level max**
This command sets the diagnostic level to its maximum.
 13. Type the command **setenv diag-switch? true**
This command enables the startup diagnostics.
 14. Type the command **reset-all**
*A message like “**Probing /pci@1f,0/pci@5 Device 8 sound**” should appear. The system is performing diagnostics. If it finds a bad device, it will print an error message. Sometimes this command can make a workstation appear to be dead. Be patient and wait about five minutes for the messages to appear. If the diagnostics seem to be freezing the workstation, power off the workstation. Then power on the workstation while holding down the STOP + N keys simultaneously. This will reset the original factory default values. Remember that the **auto-boot?** variable will be set to **true**, so the operating system will boot automatically.*
 15. Type the command **setenv diag-switch? false**
*When the variable **diag-switch?** is set to **false**, extensive hardware diagnostics will not be performed. If for some reason the diagnostics routine is preventing the server from running, press the STOP + N keys simultaneously.*
- Once all the power on self tests are performed, the OpenBoot program looks in the first track, first sector of the boot device (usually disk) for a program named **bootblk**. This program is described below.*
16. To make the operating system start, type the command **boot** at the **OK** prompt.

Phase 2 Boot Program Phase

This phase starts when the system has checked itself and starts to load the **bootblk** program from the boot device. The **bootblk** program is a small section of code on the first sector of the first track of the hard drive or tape device. When **bootblk** runs, it shows a message like

Fcode UFS Reader 1.12 00/07/17 15:48:16

bootblk only has one function. It loads the **ufsboot** program into memory and then dies. When the **Fcode UFS Reader...** message appears, **bootblk** has done its work.

The following message should now appear:

Loading: /platform/SUNW,Sun-Blade-100/ufsboot
Loading: /platform/sun4u/ufsboot

The **ufsboot** program loads the Kernel into memory. After the Kernel is loaded into memory, the **ufsboot** program dies.

It is important that a system administrator understand what is happening with the **ufsboot** program and the **bootblk** program. If the system messages shown above do not appear, the server may be dead, or something may be wrong with these two programs, which would then need to be reloaded or repaired.

Phase 3 Kernel Phase

This phase starts when the initial boot programs **bootblk** and **ufsboot** have been loaded, and the Kernel is now starting to load. The kernel can be thought of as the “core program” that defines the Solaris 9 operating system. The Kernel uses the **ufsboot** program to read Kernel modules into memory.

A Kernel module can be thought of as a dynamic piece of software code. Only the modules that are needed are loaded into the Kernel. This makes the Kernel faster and more efficient than if it always had to load all its modules into memory. After enough modules are loaded into memory, the **ufsboot** program dies.

When the front slash symbol (/) starts to swirl, the Kernel is starting to load. The SunOS Release version is now also shown. This indicates that the boot device is booting and working. If there are any further problems with the boot process, they will most likely be caused by an error in a run control script or a bad command called by a run control script.

Phase 4 The Init Phase

The init phase starts after the Kernel has loaded itself and its modules into memory. The **sched** process is the first process to be loaded. It has a PID (Process IDentification Number) of zero (0), as shown with the **ps -ef** command. The **sched** process is responsible for the scheduling policy and priority of processes. After **sched** starts up, the process called **init** is started, with a PID of one (1). The **init** process reads a text file **/etc/inittab**. Among other things, this file defines the default run level and controls how the **init** process calls up and executes run control scripts

Solaris Run Levels

Solaris 9 uses what are know as run levels. To draw an analogy, people have different age categories in life: infant, toddler, child, teen, and then adult. In the same way, Solaris goes through different run levels. The first run level loads basic device drivers. The next level loads some basic programs for local users,etc. The process continues until the default run level 3 is reached (all applications loaded and networking is enabled).

A run level can be thought of as an operating system age. For example, a system administrator could specify that at run level 2 the Oracle 9i server will start. On run level 3, users can log into the server. Each run level only allows certain programs to start. The **/etc/inittab** file controls run levels. Table 6.1 shows the different run levels possible.

Run Level	Description
S,s	Single-user system administrator state. Only the root user is allowed onto the system, from the console. Only very basic operating system services are available. File systems are unmounted unless they were mounted before the system was changed to run level S.
1	All file systems are available and all root/non-root users remain logged into the system. Only very basic operating systems services are available.
2	All root/not-root users remain logged in. Most services are available, except that the NFS (Networked File Systems) are not shared onto the network.
3	The default run level. All root/non-root users are logged into the system. The NFS file systems are shared on the network.
4	Experimental, not currently used.
5	Power-down level. All processes are shut down gracefully, all mounted file systems are unmounted and any remaining users are logged off the system. SPARC servers will power off when run level 5 is finished.
6	Server reboot.
0	Stops all system services and daemons, unmounts all file systems. It is now safe to turn off the power.

Table 6.1 Solaris Run Levels

Run Control Scripts

Solaris 9 uses what are known as "run control scripts." Often, a system administrator must create a run control script before a software package will run. For example, when the Netscape Directory server is added to a system, it does not always include the necessary run control scripts necessary to start and stop the Administration Server and the Directory Server. Netscape's instructions tell the administrator to add run control scripts with the necessary commands. It is the responsibility of the system administrator to create these run control scripts for the server.

Often, run control scripts must be customized by a system administrator. For example, one company might want to have its database program start on run level 2. Another company might want to have its accounting software start on run level 3.

Run control scripts are executed in alphanumeric order. When Solaris starts or shuts down, it executes all run control scripts in the `/etc/rc*` directories. Each directory corresponds to a run level. The directories are:

```
/etc/rc0.d  Run level 0
/etc/rc1.d  Run level 1
/etc/rc2.d  Run level 2
/etc/rc3.d  Run level 3
/etc/rcS.d  Run level S (single user mode)
```

All run control script names start with an **S** or a **K**. The scripts that start with an **S** are used to start something. For example, in the `/etc/rc3.d` directory are the scripts `S90samba` and `S50apache`. The `S90samba` script is used to start the Samba daemon `smbd` to allow the Samba service to run. The script `S50apache` is used to start the Apache webserver. The scripts that start with a **K** are used to gracefully shut something down. The script `/etc/rc1.d/K16apache` is used to kill the Apache webserver.

Understanding Run Levels

Solaris has eight run levels. Before a discussion of the `init` process can continue, it is important that the reader understand the concept of how run levels are defined in the `/etc/inittab` file.

This file contains the following critical pieces of information:

1. What processes to start, and what to do if the processes die unexpectedly
2. What to do if the system enters a new run level
3. The system's default run level

Figure 6.2 shows the contents of the `/etc/inittab` file.

```
ap::sysinit:/sbin/autopush -f /etc/iu.ap
ap::sysinit:/sbin/soconfig -f /etc/sock2path
fs::sysinit:/sbin/rcS sysinit      >/dev/msglog 2< >/dev/msglog </dev/console
is:3:initdefault:
p3:s1234:powerfail:/usr/sbin/shutdown -y -i5 -g0 >/dev/msglog 2< >/dev/msglog
sS:s:wait:/sbin/rcS                >/dev/msglog 2< >/dev/msglog </dev/console
s0:0:wait:/sbin/rc0                >/dev/msglog 2< >/dev/msglog </dev/console
s1:1:respawn:/sbin/rc1             >/dev/msglog 2< >/dev/msglog </dev/console
s2:23:wait:/sbin/rc2               >/dev/msglog 2< >/dev/msglog </dev/console
s3:3:wait:/sbin/rc3                >/dev/msglog 2< >/dev/msglog </dev/console
s5:5:wait:/sbin/rc5                >/dev/msglog 2< >/dev/msglog </dev/console
s6:6:wait:/sbin/rc6                >/dev/msglog 2< >/dev/msglog </dev/console
fw:0:wait:/sbin/uadmin 2 0         >/dev/msglog 2< >/dev/msglog </dev/console
of:5:wait:/sbin/uadmin 2 6         >/dev/msglog 2< >/dev/msglog </dev/console
rb:6:wait:/sbin/uadmin 2 1         >/dev/msglog 2< >/dev/msglog </dev/console
```

```

sc:234:respawn:/usr/lib/saf/sac -t 300
co:234:respawn:/usr/lib/saf/ttymon -g -h -p "`uname -n` console login: " -T sun
-d /dev/console -l console -m ldterm,ttcompat

```

Figure 6.2 Contents of the `/etc/inittab` file

The `/etc/inittab` file has four distinct fields. These fields are separated by the colon (`:`) symbol. The fields are:

ID:rstate:action:process

- ID** A 1- to 4-character identifier.
- rstate** Run levels that this process runs on.
- action** How to handle this process.
- process** The script or command to execute.

The **ID** and **rstate** variables need to be described in more detail.

- ID** This has a maximum of 4 characters (numbers or letters). This string is used as an identification tag by the Kernel and other processes. This value should not be changed by the system administrator. It is basically an internal book-keeping reference for the Kernel.
- rstate** When the system starts a new run level, the `init` process checks the `/etc/inittab` file. It reads each line of the `/etc/inittab` file and searches the `rstate` field. When it finds a line where the `rstate` value corresponds to the current run level, it executes the script or program associated with that line.

Examples of the `/etc/inittab` File and Run Levels

In this example, the system has just entered run level 2. The `init` process will search through the `/etc/inittab` file. Here, it has found a line with the ID `S7`.

`S7:1:respawn:/usr/lib/saf/sac -t 300`



Because the `rstate` field in this case is `1` the `init` process will ignore this line, because it is trying to find programs and scripts to execute on run level 2.

Now the `init` process reads the next line in the `/etc/inittab` file.

S39:2:respawn:/usr/lib/saf/ttymon -g -h -p

The **init** process reads this line with the ID of **S39**. It has the **rstate** field is set to **2**. It now executes the program **ttymon -g -h -p**.

S39:2:respawn:/usr/lib/saf/ttymon -g -h -p

If the process dies it will be restarted because of the **respawn** action.

The **init** process continues to go through the **/etc/inittab** file until it runs out of lines to check. It then goes to the next run level (now **3**) and scans the **/etc/inittab** files for all lines with **3** as the **rstate**. When it has checked for an executed lines with all possible run values, the **init** process stops.

Example Scripts from the **/etc/rc0.d** directory

This section illustrates some typical run control scripts in the **/etc/rc0.d** directory. The scripts present on a server can vary, depending on what software is installed on the operating system. For example, if the Apache webserver is not installed on a system, the script **S50apache** would not exist.

/etc/rc0.d

All scripts in the **/etc/rc0.d** directory are kill scripts. These scripts are used to end programs and processes. When the first of these scripts runs, it announces to users that all services are being stopped.

K00ANNOUNCE Script displays the message “System services are now being stopped”

K03samba Script used to stop Samba. Samba is a network sharing software package that lets Solaris 9 share files and printers with Microsoft Windows and other Samba-aware operating systems.

K03sshd Script used to stop the **ssh** daemon. Understand that **ssh** is used to enable secure communications between servers.

K06mipagent Script used to stop the Mobile IP agent used with mobile IP devices.

K07dmi Script used to stop the Sun Solstice Enterprise DMI service provider.

K07snmpdx Script used to kill the Solstice Enterprise Master Agent.

K10dtlogin Script used to end the CDE desktop. The desktop is then replaced by a text screen.

K16apache Script used to kill the Apache webserver.

K21dhcp One of the scripts used to stop the DHCP server. The DHCP server assigns network information to DHCP clients.

K22acct Script used to end process accounting.

Scripts in the `/etc/rc2.d` directory

- K03samba** Kills the Samba resource sharing software.
- K03sshd** Kills the SSH secure shell process.
- K21dhcp** One of the scripts used to stop the DHCP server.
- S88sendmail** Starts the `sendmail` daemon. When `sendmail` is not set up properly, the message “sendmail: unable to qualify my own name” appears. To stop these messages, change the script name to `s88sendmail`. Scripts whose names start with lower case “s” will not be run when the system starts.
- S99dtlogin** Starts the Login Manager. That is the login GUI screen that is seen after the system boots up.

Lesson 6.2 How to Work with Run Control Scripts

If an `/etc/rc*` script becomes damaged, it could disable a server when the system tries to gracefully shut down or start. These steps show how to safely work with these scripts.

1. Log in as the root user.
2. Open a Terminal window.
To open a terminal window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Tools menu item, then left click on the Terminal window icon.
3. Type the command `cd /etc/rc3.d`
The command `cd /etc/rc3.d` switches to the `rc3.d` directory.
4. Type the command `echo "echo \"my script\"" > S99zzz`
The command `echo "echo \"my script\"" > S99zzz` creates a shell script that causes the system to display the message `my script` when the system starts. (You could also create this script by running `vi` and creating the `S99zzz` file with the single line `echo "my script"` in it.)
5. Type the command `cat S99zzz`
*The script should only have **one** line that reads exactly as shown here:
echo "my script"
*If this is not true, repeat the previous step and then continue with this one.**
6. Type the command `./S99zzz`
This tells the system to run the script that was just created.
7. Notice that the error message `./S99zzz: cannot execute` appears, and the `S99zzz` script does not run. If this happens when the server starts or reboots, the server would freeze up. **Before starting or rebooting a server or workstation with a new run control script, try the script from the command line to make sure it does not hang.** If a run control script hangs during the boot process the server could become unstable. It is crucial that the system administrator try the run control script before the reboot.
8. Type the command `chmod 777 S99zzz`
*The command `chmod 777` makes the script executable. In a production environment `chmod 777` would never be used on a run control script, but it will work for purposes of this lesson. **For highly secure systems, always make a script executable by setting its permissions to match the company's server security guidelines.** The command `chmod 744` creates the standard permission on a run control script.*
9. Type the command `./S99zzz`
*Now, the script should run properly and should display the text `my script` on the screen. If the script does not run properly, check the previous commands. The key point is to make a text script with just one line that reads:
echo "my script"*
10. Type the command `init 6`
The command `init 6` reboots the server. Look for the line `my script` when the server starts again.

11. Log in as the root user.
12. Type the command `cd /etc/rc3.d`
This switches to the directory /etc/rc3.d
13. Type the command `mv S999zzz s999zzz`
*The `mv` command can be used to change a file's name. In this case, the file named `S999zzz` is renamed to `s999zzz`. Remember that Solaris 9 ignores any run control script whose name does not start with a capital `S` or a capital `K`. Because of this, a safe way to disable a script is to change the first letter to a lower case `k` or lower case `s`. **Do not delete a run control script.** Understand that run control scripts take up almost no disk space. If they are deleted, they have to be restored from a tape backup. Solaris 9 does not have a mechanism to **undelete** a file. Once a file is deleted, it is gone. The "Trash Can" utility does not typically work with command line tools.*
14. Type the command `cd /etc/rc3.d`
15. To see a typical run control script, type the command `more S77dmi`
The command `more` displays text files one screen at a time. The `more` command can not damage a file so it is a safe command to look at run control scripts.
16. Type the command `exit`
When the command `exit` is typed, it exits the user's shell (Bourne, C, Korn, etc...). If the last open shell is deleted, the Terminal window closes.

Key Points to Remember

It is important that a reader understand how the boot process works in Solaris 9, and to understand the `/etc/inittab` file. There are plenty of situations in which a software package requires the system administrator to add a run control script. Remember that run control scripts must be made executable after they are created, and that you should always test new run control scripts at the command line before restarting the system.

Chapter 7 Working with Users and Groups

Lessons in This Chapter

Lesson 7.1 Creating Users with Command Line Tools	7-5
Lesson 7.2 Modifying a User with the <code>usermod</code> Command	7-8
Lesson 7.3 Removing a User with the <code>userdel</code> Command	7-8
Lesson 7.4 Using <code>chmod</code> and <code>ls -l</code>	7-10
Lesson 7.5 Setting File Permissions	7-13
Lesson 7.6 Creating a group From the Command Line	7-15
Lesson 7.7 Removing a Group with the <code>groupdel</code> Command	7-16
Lesson 7.8 Using the Admintool to Create a User Account	7-16
Lesson 7.9 Solaris User Manager 1.4	7-18
Lesson 7.10 Create Users from a Template	7-24
Lesson 7.11 Setting Up a Quota	7-31
Lesson 7.12 The <code>repquota</code> Command	7-34
Lesson 7.13 Changing a User's Quota Limit	7-35
Lesson 7.14 Disabling a File System's Quota Limit	7-36

Introduction

What is a Solaris user? What is a Solaris group? Every person that would like to work on a Sun Solaris server or workstation must log in as a user. A user account has a username (also known as a login name) and a password. The username and password provide system security and system auditing. If a user acts in a malicious manner, that user's conduct can be traced through various security tools within Solaris and third-party security tools.

A user's account usually has a home directory, but this is not absolutely required. A home directory has configuration files that relate directly to a user. If a user prefers a specific type of font when he or she uses the CDE, that user's desktop settings are stored in a home directory. The home directory is also where a user has full permission to create, modify, and delete files and directories. The home directory can be thought of as a user's *piece of real-estate* on the server. Typically the user's home directory is named `/usr/<username>` or `/export/home/<username>`. The location of a user's home directory can be customized to almost any directory on the server.

File and directory permissions are set by user accounts. For example, a system administrator can create a directory named `/mydirectory` and only allow a user `janderson` the right to read files from that directory. Actually, Solaris uses numbers instead of names, so each username has a UID (User IDentification) number associated with it. A UID number acts just like a social security number for a person. The UID is used by Solaris 9 to set file and directory permissions.

After the user account is created, a quota can be set up on the user's account. This quota will limit the amount of disk space that a user is allowed to have. If one user created a 10 gigabyte file (10 GB), that file could easily fill up the file system dedicated to user's home directories. Using a quota, a user can not use too much disk space. There are also only so many files that a file system can contain. A quota can also be set up to limit the number of files a user can create. User quotas are essential for an ISP (Internet Service Provider) that uses Sun Solaris for its customers' Internet accounts.

Solaris 9 features several command line utilities and a GUI utility named Admintool to create and administer users and groups. The Admintool is scheduled to be phased out of Solaris 9, so do not be too surprised if it is not included in later releases.

Files Related to Users

There are three main files in the `/etc` directory that define a user. The three files are `/etc/passwd`, `/etc/shadow` and `/etc/group`. There are other files related to a user, but for now, just understand that these are the three primary files.

The `/etc/passwd` file

The `/etc/passwd` file is the primary file that defines a user. If a username does not exist in this file, the user does not exist. This is a text file that can be modified directly by the system administrator with a text editor, or it can be modified by the various utilities used to administer a user. Figure 7.1 shows what the text inside a typical `/etc/passwd` file looks like.

```
janderson:x:231:143:John Anderson:/usr/jandersonhome:/bin/sh  
ksmith:x:203:147:Karen Smith:/usr/kmit:/bin/csh
```

Figure 7. 1 The `/etc/passwd` File

To demonstrate the fields used in the `/etc/passwd` file, John Anderson's account and Karen Smith's account will be used as an example. Notice that the fields in the `/etc/passwd` file are separated by a colon (:).

The fields are:

- **loginID** – This is the username (also called **user login account**, **user account**, **login account**, **loginID**, or **userID**) that the user types when he or she wants to access the system. In this example, John Anderson would type **janderson** at the login screen to access his account. Karen Smith would type **ksmith**. (As shown in Figure 7.1, John Anderson = janderson and Karen Smith = ksmith.)
- **x** – The **x** after everyone's loginID is a blank password holder. For security reasons the password is not printed in clear text anywhere in the system. This **x** informs the operating system that there is an encrypted password associated with this user. The operating system then checks the `/etc/shadow` file for the user's password. The `/etc/shadow` file will be described in greater detail later in the chapter.
- **UID** (User ID) – The UID is a numeric field that is used to describe a user. Just like people in the United States have a social security number; users in a Solaris system have a UID that numerically identifies them. In this example John Anderson's UID is 231 and Karen Smith's ID is 203.
- **GID** (Group ID) – The GID is the number of the primary group that a user belongs to. Members of a group can share files and resources. Just as users are represented by a name and a number, groups are represented by a name and a number. The file `/etc/group` has the group name that corresponds to the GID. Every user must belong to one primary group. A user can also belong to additional groups, but that is optional. Groups can be something like "Sales" or "Marketing" or "Engineering." These groups can then be used to set permissions on files and directories. For now, just know that the `/etc/passwd` file defines the user's primary group. Let's say that the sales team has a GID of 143 and the marketing team has a GID of 147. In this example John Anderson is a member of the sales team (143) and Karen Smith is a member of the marketing team (147). If a directory only has read permission for the sales team (143), John can read (but not write) files from the directory, while Karen can neither read nor write files from it.

- **Comment** – This is a text section that allows the administrator to add a comment like “John M. Anderson” or “Guest User Account” so that he or she will know what the account is referencing. In this example John Anderson’s name and Karen Smith’s name are in the comment field. The administrator can put any comment in this field. For example, the comment “Temp Account - Delete May 12” is a valid comment.
- **Home-Directory** – This indicates where a user’s home directory is located. A home directory stores the user’s configuration files, text files, database files or any other files the user would like to store in his or her home directory. The home directory also saves some system configuration information about this user. A home directory is not a requirement for a user, so this field can be blank. In this example John Anderson’s home directory is **/usr/jandersonhome** and Karen Smith’s home directory is **/usr/kmit**.
- **Login-shell** – This indicates the user’s shell. Unlike Microsoft Windows, Solaris allows a user to select what kind of operating system he or she prefers. This selection is known as a “shell.” For example, some users might like the way the “C shell” operates and can request the “C shell” from the system administrator. Others might like the “Bourne” shell and will request this from the system administrator. This field in the **/etc/passwd** file defines what shell the user would like to use. In this example John Anderson has the Bourne shell and Karen Smith has the C shell.

The **/etc/group** file

The **/etc/group** file is another critical file for users. It defines the system groups. This is a text file that the system administrator can change with a text editor or modify with command line utilities. The Admintool and the SMC (Solaris Management Console) can also modify this file.

Groups are useful when it comes to setting up user permissions throughout the system. Imagine if a system administrator had thirty people in the engineering department, and the engineering manager wanted to give these thirty people permission to a directory. It would be very tedious to assign permissions to all thirty people one by one. If a group is created, all thirty people can be added to the group, and then permissions can be set via the group. Figure 7.2 shows a section of a typical **/etc/group** file.

```
Sales:x:143:janderson
Marketing:x:147:ksmith, mpeterson
```

Figure 7. 2 The **/etc/group file**

The **/etc/group** file holds four key pieces of information. The fields in the **/etc/group** file are separated by colons (:).

These fields are:

- **Groupname** – This is similar to the loginID. It is a name that has a meaning, such as "Sales" or "Marketing" or "Distribution." In this example there are two groups: Sales and Marketing.
- **Group-password** – This does not exist any more. However, because some older programs might try to reference this field, it must be included. The **x** shown here is just a legacy placeholder.
- **GID** – This is a numeric identifier for the group, somewhat like a social security number. In this example **143** identifies the **Sales** group and **147** identifies the **Marketing** group.
- **Usernames** – This is a list of the users that are members of the group. If there is more than one person in a group, the usernames are separated by commas (.). The names must correspond to the loginID names in the

`/etc/passwd` file. In this example, the Sales group has **janderson** and the Marketing group has **ksmith** and **mpeterson**.

The `/etc/shadow` file

The `/etc/shadow` file is a text file that holds the encrypted password for each user, and the password aging associated with the user. A system administrator does not need to know how the encryption is done, just that the encrypted password is stored here. The `/etc/shadow` file also defines how long a password can exist before it must be changed, what is the shortest period of time a password can exist before a user can change the password, and how many days before a password expires the system should warn the user about this. The `/etc/shadow` file also contains one last piece of information, the date when the user last changed the password.

Figure 7.3 shows a section of a typical `/etc/shadow` file.

```
janderson:@Ndx8DNZS^DNz:11928::::::  
ksmith:*LK*:11928::::::
```

Figure 7.3 The `/etc/shadow` File

The fields are:

- **LoginID** – This is the same loginID as in first field of the `/etc/passwd` file. In this example **janderson** and **ksmith** both have loginIDs in this file.
- **Password** – An encrypted version of the user’s password. It can be a combination of 13 characters that is the encrypted password. If a user's account is locked, this will be ***LK***. If no password has ever been set for the user, this will be **NP**.
- **Lastchanged** – This is the last time the password was changed, referenced as an absolute number of days after 01/01/1970.
- **Min** – The minimum number of days that password can be used before it can be changed.
- **Maximum** – The maximum number of days that a password can be used before it must be changed.
- **Warn** – The number of days ahead that a user is warned to change his or her password before it expires.
- **Inactive-days** – The number of days the account can go unused before it is locked (prevents stale accounts.)
- **Expire** – The expiration date, after which the account will not work.

Notice that in the example lines above, the last five fields are blank. This indicates that these values have never been set for the users shown.

A Closer Look at Solaris Users

The previous sections described the various components that create a user. In the following section; we will examine in greater detail some of the elements that make a Solaris user. Additional material will build upon the previous material.

The Login Shell

Microsoft Windows has only one command interpreter (the MS-DOS window). But Solaris lets users choose among different command interpreters, and the commands that go with them. Solaris uses what is known as a **shell** to present an interface to the user and to interpret what the user wants to do (such as copy files or move to a different directory). The shell then *talks* to the Kernel in machine language instructions (such as move file 2164 to location 2013, change to directory 3252, or start process 246). A user can choose from three different shells: the Bourne Shell, the Korn Shell and the C Shell. These shells are not described in detail here; just know that the users can choose a shell to use.

The Initialization File

Each user has an initialization file that contains specific settings that apply to the user. Some users might want a larger screen font, to have an application loaded that requires that certain system variables be set, or to have a local printer be their default printer. The initialization shells save this information. These files are hidden because they all start with a period (.) as the first character of the file. Most inexperienced users can not see these files and will not know they exist. Power users and experienced users can still view and modify these files if they want to modify these settings.

Bourne Shell	.profile
Korn Shell	.profile
C Shell	.login .cshrc

Creating User Accounts

Solaris 9 requires each user to log onto the system before he or she can use it. The user accounts can be created with command line utilities or by using a GUI tool known as the **Admintool**. The command line utilities are

groupadd	Command line tool to add a group to the system
useradd	Command line tool to add a user account to the system
usermod	Command line tool to modify a current user account
userdel	Command line tool to remove a user account

Lesson 7.1 Creating Users with Command Line Tools

In this lesson a simple user is created, named **myuser**. After the user account is created, the various configuration files that relate to the user are examined to see how a new user account is added to the system.

1. Log in as the root user.
To create a user's account with a command line utility you must be a root user (any account with a UID = 0).
2. Open a Terminal window.
To open a Terminal window, right click anywhere in unoccupied desktop space. When the Workspace menu appears, left click on the Tools menu item, then left click on the Terminal icon.
3. Type the command **groupadd mygroup**
*The **groupadd** command creates a new group in the **/etc/group** file.*
4. Type the following command all on one line (*do not press the Return key until the very end*): **useradd -g mygroup -m -d /usr/myuser -c "account I created" -s /bin/sh -u 203 myuser**
The command just typed does the following things.
useradd (runs the program)

-g mygroup (adds the user to the group named “mygroup”)
-m (makes a home directory for this user)
-d /home/myuser (specifies the location of the home directory)
-c “account I created” (adds the comment “account I created” in the **/etc/passwd** file)
-s /bin/sh (selects the Bourne shell script)
-u 203 myuser (specifies the userID of 203; ordinarily, the **useradd** command will generate this automatically.
myuser (loginID of the user)

5. Type the following command: **cat /etc/passwd**
 Try to find the changes that were created by the **useradd** command above for the **myuser** account?
6. Type the following command: **cat /etc/group**
 Where is the group named **mygroup** in the file?
7. Type the following command: **cat /etc/shadow**
 Try to find the account **myuser** in this file. Some users may not have a password set. The word **NP** indicates a password has not been set for the user. The word ***LK*** shows that an account has been locked..
8. Type the command: **passwd myuser**
 The **passwd** command is used to change a user’s password.
9. Type the password **my123**
10. Retype the password **my123**
11. Type the command **cp /usr/myuser/local.profile /usr/myuser/.profile**
 This copies the user’s **local.profile** file to a hidden file called **.profile**, which the user cannot easily access. When a filename has a period (.) at the start, it is a hidden file.
12. Type the command **rm /usr/myuser/local.profile**
 This uses the **rm** command to remove (delete) the file that was just copied.
13. Type the command **cat /etc/shadow**
 Try to find the account **myuser** in this file. Does it have an encrypted password now? Does it have **NP** in one of the fields? Remember, the word **NP** indicates a password has not been set for the user. In this case the **myuser** account has an encrypted password.
14. Log off as the root user and log back in as **myuser** with the password **my123**.
15. Type the command **pwd** to see the present working directory.
 This is the home directory of the **myuser** account.
16. Type the command **ls -Al**
 Is there a file named **.profile**? This is the Bourne shell script’s initialization file.
17. Type the command **cat .profile**
 These are the user’s settings that apply to the **myuser** account.

A Closer Look at the **/etc/passwd**, **/etc/group** and **/etc/shadow** Files

The **/etc/passwd** file

As mentioned above, the fields in the **/etc/passwd** file are:

- **LoginID** – This LoginID must start with a letter and must contain at least one lower case character. It can contain any combination of numbers, letters, and the period (.), hyphen (-) and underscore character (_). It can not be longer than eight characters or shorter than two characters.
- **x** – This is a throwback password holder and is always an **x**. The Solaris 9 operating system checks the **/etc/shadow** file for the user’s password.
- **UID** – This is a numeric field that represents a user. There are some user accounts that have special meaning to the Solaris operating system. For example, the root user has a UID of zero (0). Solaris 9 has some system

accounts that it uses internally. Nobody can log in through these accounts. Table 7.1 shows the range of user IDs and the restrictions on these numbers.

UID Range	Type of Accounts
0	Root user.
1 – 100	Internal system accounts.
101-60000	Any user account can use these.
60001	This is used with a root request from an NFS server. This is all the details needed for this certification material.
60002	Accounts that access a system application by a user or process that is not logged onto the system.
65534	A throwback “nobody” account from previous versions of Sun Solaris.

Table 7.1 UID Numbers

- **GID** – This is a numeric field that indicates what primary group a user belongs to. Groups can be something like **sales, marketing, engineering**, and so forth. The groups can then be used to set permissions and other system tasks for multiple users. Groups 0 – 99 are internal system groups that users can not access. Group numbers 100 – 60000 are accessible groups to users.
- **Comment** –The administrator can add a comment like “**John M. Anderson**”
- **Home-Directory** – The full path to the user’s home directory.
- **Login-shell** – The user’s default shell. The three standard shells that come with Sun Solaris are **/bin/sh** (Bourne Shell) **/bin/ksh** (Korn Shell) and **/bin/csh** (C shell).

The /etc/group File

The fields in the **/etc/group** file are:

- **Groupname** – This is similar to the loginID. It is a name that has meaning to a person such as **Sales, Marketing, or Distribution**.
- **Group-password** – This does not exist any more; it is left in as a legacy placeholder
- **GID** – A GID acts just like a UID, except that it represents a group. The GID number ranges are described in Table 7.2.

GID	Type of Group
0	Anyone in this group is a root user.
0-99	Reserved for internal system usage.
14	Sysadmin group; anyone in this group can use the Admintool just like a root user.
100 – 60000	Groups anyone can use.,
60001 and 60002	Reserved for internal system usage.

Table 7.2 GID Numbers

- **U usernames** – This is a list of the users that are members of the group. The names correspond to the loginID names that are found in the `/etc/passwd` account

The `/etc/shadow` File

The fields in the `/etc/shadow` file are:

- **LoginID** – This is the same loginID as in first field of the `/etc/passwd` account.
- **Password** – Has an encryption of the user’s password. It can be a combination of 13 characters that is the encrypted password, or ***LK*** – account locked, or **NP** – no password.
- **Lastchanged** – the last time the password was changed, this is referenced from 01/01/1970 as an absolute date in days.
- **Min** – minimum number off days a password can be used before it can be changed.
- **Maximum** – maximum number of days a password can be used before it must be changed.
- **Warn** – number of days a user is warned to change his or her password before it expires.
- **Inactive-days** – number of days the account can go unused before it is locked (prevents stale accounts.)
- **Expire** – absolute expiration date, the account will not work after this date.

Lesson 7.2 Modifying a User with the `usermod` Command

The `usermod` command is useful in that it allows the system administrator to modify existing user accounts. After you use `usermod`, if for some reason the changes to the user’s account do not seem to have taken effect, have the user log off and then log on again.

1. Log in as the root user.
2. Open a Terminal window.
To open a Terminal window, right click anywhere in unoccupied desktop space. When the Workspace menu appears, left click on Tools menu item, then left click on the Terminal icon.
3. Type the command `usermod -l myuser2 myuser`
This command changes the username `myuser` to `myuser2`.
4. Log off the system.
5. Login as `myuser2` with the password `my123`
6. Log off the system
7. Log back in as the root user.
This Lesson demonstrated how to use the `usermod` command line utility to change the username `myuser` account to `myuser2`.

Lesson 7.3 Removing a User with the `userdel` Command

The `userdel` command is used to remove a user from the system. The `userdel` command can be instructed to remove the user’s home directory. This command will not remove files that the user created in other file systems. For example, if the user created a big file in the `/tmp` directory, the `userdel` command will not detect or clean that file off the system.

1. Log in as the root user

2. Open a Terminal window.
To open a Terminal window, right click anywhere in unoccupied desktop space. When the Workspace menu appears, left click on Tools, then left click on the Terminal icon.
3. Type the command **userdel -r myuser2**
*This command removes the **myuser2** user and removes the home directory associated with **myuser2**. If anyone tries to log in as **myuser2** he or she will not succeed, as it does not exist anymore.*

*If the **myuser2** account created files on the server outside the user2 home directory, these files will not automatically be deleted. For example, if **myuser2** created a file in the **/tmp** directory, that file will still exist after the **userdel** command is run. The **userdel** command does not clean out all files that **myuser2** created.*

Permissions on Files and Directories with Users

Files and directories can have read, write and execute permissions. These permissions are important because access to files or directories need to be restricted between users. Imagine if a file named **termination-list** could be viewed by any user on the server! The only user exempt from the read/write/execute permissions on a file or directory is the root user.

Directory Permissions

When a user creates a directory, the user can set read, write and execute permissions on it with the **chmod** command. The read/write/execute permission on a directory allows or restricts other users from reading text files, writing text files, or executing programs in the directory. These permissions are used for security purposes so that files and directory contents can be kept confidential from other users. A user can even restrict himself or herself from reading, writing or executing files in a directory (this protects those files from accidentally being deleted by the user). Only the root user can change permissions on system directories like **/usr**, **/etc**, and **/opt** and other directories that hold system-critical files.

Directory Read Permission

If a user does not have read permission for a directory, the user can not look into the directory or move into it. If a user does have read permission for a directory, the user can look at the contents of the directory and can move into it. With read permission, the user can copy files and sub-directories (depending on their read permissions) out of the directory.

Directory Write Permission

If a user has write permission for a directory, the user can create files and sub-directories in it.

Directory Execute Permission

If a user has execute permission for a directory, the user can run any program or script in the directory.

The read/write permissions can be set up so that other users who belong to the same group can read, write or execute a file, while users outside the group can not read, write or execute files in the directory.

The only exception to these rules is the root user. The root user is “all powerful” in that it can read, write or execute any file in any directory. Users cannot deny the root user access to any part of the server.

File Permissions

When a user creates a file, the user can set read, write and execute permissions on the file with the **chmod** command. The read/write/execute permission on a file allows or restricts other users from reading text files (this

includes copying a text file) writing text files, or executing program files. A user can even restrict himself or herself from reading, writing or executing files created by the user.

File Read Permission With this permission, a user can read the contents of a text file and can copy it. The user can not run a shell script or a program with just read permission.

File Write Permission With this permission, a user can modify a text file. The user can also write over the original file with a new file.

File Execute Permission With this permission, a user can execute a shell script or program. The user does not have permission to look into a file with only execute permission. This permission does not allow a user to modify a file..

The chmod Command

The **chmod** command is used to set the permissions on a file or directory. It uses numbers to set permissions, as follows:

- 7 - read/write/execute
- 6 - read/write
- 5 - read/execute
- 4 - read
- 3 - write/execute
- 2 - write
- 1 - execute
- 0 - no permissions

The permissions are set by the **chmod** command for the user himself, users in the same group and for all other users. Table 7.3 shows some chmod permissions.

chmod Permission	Permissions Set	Character Listing
chmod 744 file	user rwx group r-- others r--	rwx r-- r--
chmod 317 directory	user -wx group --x others rwx	-wx --x rwx
chmod 215 file	user -w- group --x others r-x	-w- --x r-x
chmod 777 file	user rwx group rwx others rwx	Rwxrwxrwx

Table 7.3 Chmod Examples

Understand that file and directory permissions are cumulative. If a user has read permission in any category (user, group, others) that user can read the file. All it takes is one valid **r**, **w** or **x** permission in any category, and the user is granted permission to perform the appropriate function.

When the command **ls -l** (small **L**) is used, it shows what is known as a “long listing” of files and directories. This command shows what permissions exist on the file or the directory.

Lesson 7.4 Using chmod and ls -l

This lesson uses the **chmod** command to set permissions on a file and a directory. Then, the **ls -l** command is used to view the directory’s or file’s permissions. In this lesson, two users are created, **user11** and **user22** (if they don’t already exist). The **user11** user creates files and directories. After **user11** creates the files and directories, **user22** tries to read and write files in **user11**’s directory.

1. Log in as the root user
2. Open a Terminal window
To open a Terminal window, right click anywhere in unoccupied desktop space. The Workspace menu appears. Left click on the Tools menu item, then left click on the Terminal icon.

If user11 and user22 do not exist, type the following commands:

```
useradd -m -d /usr/user11 user11
cp /usr/user11/local.profile /usr/user11/.profile
passwd user11 (set user11's password to user11)
useradd -m -d /usr/user22 user22
cp /usr/user22/local.profile /usr/user22/.profile
passwd user22 (set user22's password to user22)
```

These commands were used in lesson 7.1. They are only being used here to create the users for this lesson. Read Lesson 7.1 and its accompanying text for more information on these commands.

3. Exit out of the CDE.
4. Login as **user11**.
5. Open a Terminal window.
To open a Terminal window in the CDE, right click anywhere in unoccupied desktop space. Left click on Tools, then left click on the Terminal icon.
6. Adjust this Terminal window so that it occupies only the right half of the desktop.
7. Open another Terminal window.
8. Adjust the second Terminal window so that it occupies the left half of the desktop.
9. Type the command **su - user22**
*The **su - <user>** command changes one user into another user. Do not worry about how the **su -** command works; just understand that **user22** is in the new Terminal window.*
10. Adjust the Terminal window so that it occupies the right half of the desktop.
11. Type the command **id** in both Terminal windows.
The desktop should now look like Figure 7.4.

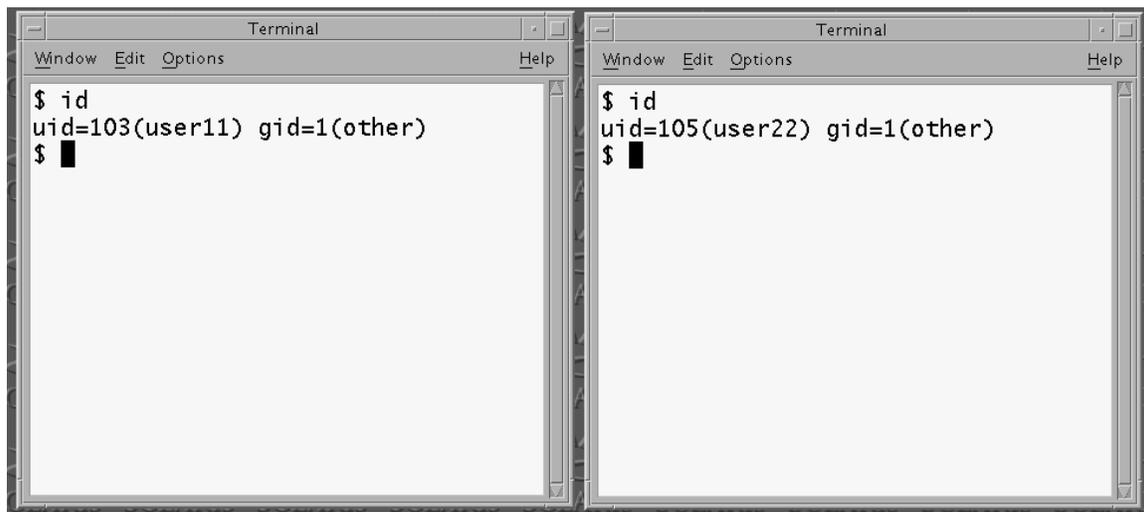


Figure 7. 4 Terminal Windows

*The command **id** shows the following items: UID (username) GID (groupname). The UID is like a social security number for a person. The username is the login name. The GID is a numerical representation of the*

group the user belongs to. The groupname is the name of the group the user belongs to. Notice that both **user11** and **user22** are members of the group named "other" with the GID=1.

12. In the left Terminal window (**user11**) type the command **pwd**
13. In the right Terminal window (**user22**) type the command **cd /usr/user11**
*The command **cd /usr/user11** changes user22's current working directory to the **/usr/user11** directory. Now **user22** is in **user11**'s home directory.*
14. In the right Terminal window (**user22**) type the command **pwd**
*The command **pwd** shows what directory **user22** is currently in (**/usr/user11**)*
15. In the left Terminal window (**user11**) type the command **mkdir user11dir**
*The command **mkdir user11dir** makes a directory named **user11dir**.*
16. In the left Terminal window (**user11**) type the command **ls -l user11dir**
*The command **ls -l** is used to show the permissions on a directory. The output of the **ls -l** command should be:*

```
drwxr-xr-x  2 user11  other          512 Jun 20 15:41 user11dir
```

The permissions are as follows

d(signifies a directory)**rw**x(user)**r-x**(group)**r-x**(others)

The permissions are as follows:

- The first character **d** – shows that this is a directory.
- The next 3 characters **rw**x show that the user has read/write/execute permission to the directory.
- The next 3 characters **r-x** show that the user's group has read and execute permission to the directory.
- The next 3 characters **r-x** show that all other users have read and execute permission to the directory.

17. In the left Terminal window (**user11**) type the command **chmod 700 user11dir**

*The command **chmod 700** sets permissions on the directory as follows:*

7: Allows **user11** to read, write, and execute all files in the directory.

0: The first zero sets the **group** permissions to none. Other members of **user11**'s group cannot read, write, or execute any files in this directory.

0: The second zero sets permission for **other users** to none. Users who are not in **user11**'s group cannot read, write, or execute any files in this directory.

18. In the left Terminal window (**user11**) type the command **ls -l user11dir**
*The command **ls -l** is used to show the permissions on a directory. The output of the **ls -l** command should be:*

```
drwx-----  2 user11  other          512 Jun 20 15:41 user11dir
```

The permissions are as follows:

- The first character **d** – shows that this is a directory.
- The next 3 characters **rw**x show that the user has read/write/execute permission to the directory.
- The next 3 characters **---** show that the user's group has no permissions to the directory.
- The next 3 characters **---** show that all other users no permissions to the directory.

19. In the right Terminal window (**user22**) type the command **cd user11dir**
20. There should be an error message: **user11dir: permission denied**
*As shown above, **user22** can not enter into the **user11dir** directory because **user22** does not have permission to enter it. As shown above, **User11** set the permissions to **700**. The first zero (**0**) is for groups (no permission) and the second zero (**0**) is for others (no permissions)*

21. In the left Terminal window (**user11**) type the command **chmod 770 user11dir**
22. In the left Terminal window (**user11**) type the command **ls -l user11dir**
*The command **ls -l** is used to show the permissions on a directory. The output of the **ls -l** command should be:*

```
drwxrwx---  2 user11  other          512 Jun 20 15:41 user11dir
```

The permissions are as follows:

- *The first character **d** – shows that this is a directory.*
 - *The next 3 characters **rwx** show that the user has read/write/execute permission to the directory.*
 - *The next 3 characters **rwx** show that the user’s group has read/write/execute permission to the directory.*
 - *The next 3 characters **---** show that all other users no permissions to the directory.*
23. In the right Terminal Window (**user22**) type the command **cd user11dir**
 24. In the right Terminal window (**user22**) type the command **pwd**
*As can be seen with the **pwd** command, **user22** made it into the **user11dir** directory because **user11** set the group permissions to **770**. The first **7** gave **user11** the permissions (read, write, execute), the second **7** gave others in the same group this permission (**user11** and **user22** are in the same group) the permissions, and the **0** gave all other users no permissions.*
 25. Try playing with the **chmod** command, the **ls -l** command, the **cd** command and the **pwd** command. This is the best way to learn how directory permissions work.

Do not close down these Terminal windows! They will be used in other examples.

Lesson 7.5 Setting File Permissions

In this lesson, readers are going to create a file as **user11** and then will have **user22** try to add text to this file. Permissions are going to be set on the file that will enable or disable file access to **user22**.

Lesson 7.4 describes how to setup the Terminal windows for this lesson and future lessons.

1. In the left Terminal window (**user11**) type the command **pwd**
*This shows the current working directory for **user11**.*
2. In the right Terminal window (**user22**) type the command **cd /usr/user11**
*The command **cd /usr/user11** changes **user22**’s current working directory to the **/usr/user11** directory. Now **user22** is in **user11**’s home directory.*
3. In the right Terminal window (**user22**) type the command **pwd**
*This confirms that **user22** is currently in **user11**’s home directory (**/usr/user11**).*
4. In the left Terminal window (**user11**) type the command
echo "file made by user11" > user11file
*The command **echo "file made by user11" > user11file** creates a text file named **user11file** in **user11**’s home directory.*
5. In the left Terminal window (**user11**) type the command **cat user11file**
*The command **cat user11file** displays the text inside the text file **user11file**.*
6. In the left Terminal window (**user11**) type the command **ls -l user11file**
*The command **ls -l** shows the permissions on a file or directory. The output of the **ls -l** command should be:*

```
-rw-r--r--  2 user11  other          512 Jun 20 15:41 user11file
```

The permissions are as follows:

- The first dash `-` shows that this is a file.
- The next 3 characters `rw-` show that the user has read/write permission to the file.
- The next 3 characters `r--` show that the user's group only has read permission to the file.
- The next 3 characters `r--` show that all other users only have read permission to the file.

7. In the right Terminal window (**user22**) type the command `cat user11file`
As seen, **user22** can read the **user11file** text file because **user11** has permission to read the file. Both **user11** and **user22** are members of the same group.
8. In the left Terminal window (**user11**) type the command `chmod 700 user11file`
9. In the right Terminal window (**user22**) type the command `cat user11file`
The command `cat user11file` no longer works, because **user11** changed the permissions on the file to **700**.
10. In the left Terminal window (**user11**) type the `ls -l` command. The output of the `ls -l` command should be:

```
-rwx----- 2 user11 other 512 Jun 20 15:41 user11file
```

The permissions are as follows:

- The first dash `-` shows that this is a file.
- The next 3 characters `rwx` show that the user has read/write/execute permission to the file.
- The next 3 characters `---` show that the user's group only has no permissions to the file.
- The next 3 characters `---` show that all other users only have no permissions to the file.

11. In the left Terminal window (**user11**) type the command `chmod 777 user11file`
The command `chmod 777` sets all permissions (user, group, others), to **7**. This means that all users have read, write, execute permission for **user11file**
12. In the right Terminal window (**user22**) type the command `cat user11file`
The command `cat user11file` works, because **user11** changed the permissions on the file to **777**.
13. In the left Terminal window (**user11**) type `ls -l` command.
The output of the `ls -l` command should be:

```
-rwxrwxrwx 2 user11 other 512 Jun 20 15:41 user11file
```

The permissions are as follows:

- The first dash `-` shows that this is a file.
- The next 3 characters `rwx` show that the user has read/write/execute permission to the file.
- The next 3 characters `rwx` show that the user's group has read/write/execute permission to the file.
- The next 3 characters `rwx` show that all other users have read/write/execute permission to the file.

14. In the right Terminal window (**user22**) type the command `echo "line added by user22" >> user11file`
The command `echo "line added by user22" >> user11file` appends the text "line added by user22" at the end of **user11file** ("`>>`" is the append operator). This is possible because the group permission on the file is `rwx`. The `w` means that anyone in the same group can write to the file.
15. In either the right or left terminal (**user22**) or (**user11**) type the command `cat user11file`
16. In the left Terminal window (**user11**) type the command `chmod 700 user11file`
This sets the group permissions to **0**.
17. In the right Terminal window (**user22**) type the command `echo "try to add another line by user22" >> user11file`
The command `echo "try to add another line by user22" >> user11file` failed because the group permission on the file is `---`. The `---` means that nobody in the same group can write to the file.

Play with the **chmod**, **ls**, **mkdir** and **touch** commands. Practice setting permissions and changing files and directories.

Using the **u g and o** Options with **chmod**

Instead of using numbers, the letters **r w x** and **u g o** can be used to set permissions, as follows:

r=read, **w**=write, **x**=execute, **u**=user, **g**=group, and **o**=others

Here are some examples:

- | | |
|--------------------------------|---|
| chmod u=x | Sets the permissions u = --x (g = not affected , o = not affected). The group and others permissions are not changed by this command. |
| chmod u+x | Adds the permission u=x to whatever the permissions were before. |
| chmod u-x | Removes the permissions u=x from whatever the permissions were before. |
| chmod u=rw,g=rwx | Sets the permissions u= rw- g= rwx (o= not affected). |
| chmod o-rx | Removes the read and execute permissions from others (u = not affected, g = not affected). |
| chmod u=rwx,g=rwx,o=rwx | sets the permissions u=rwx g=rwx o=rwx. |

Groups Within Solaris 9

As we've seen earlier, Solaris 9 lets the system administrator create groups, or collections of user accounts. For example, a system administrator can create groups named **Sales**, **Marketing**, and **Engineering**. After the groups are created, the system administrator can assign users to each of the groups.

System administrators like groups because groups make the job of assigning file and directory permissions much easier. Imagine if a new engineering file was created and the system administrator had to assign read permissions to this file for everyone in the sales department user by user! Obviously, groups are a necessity for a large company. Just like users have a UID, groups have a GID (Group IDentification). Groups are known by a group name and a group ID number.

Lesson 7.6 Creating a group From the Command Line

In this lesson the **groupadd** command will be used to create a group named **mygroup2**. The **/etc/group** file is then examined with the **cat** command.

1. Log in as the root user
2. Open a Terminal window
To open a Terminal window, right click anywhere in unoccupied desktop space, left click on the Tools menu item, then left click on the Terminal icon.
3. Type the command **groupadd -g 279 mygroup2**
*The command **groupadd** uses the **-g** option to specify a group number of 279. The name of the group is **mygroup2**.*
4. Type the command **cat /etc/group**
*The **cat** command shows the group named **mygroup2** with a groupID of 279 .*

Lesson 7.7 Removing a Group with the `groupdel` Command

The `groupdel` command is used in this lesson to remove the `mygroup2` group created in the previous lesson.

1. Log in as the root user
2. Open a Terminal window
To open a Terminal window, right click anywhere in unoccupied desktop space, left click on Tools, then left click on the Terminal icon.
3. Type the command `groupdel mygroup2`
This command removes the `mygroup2` group from the system.
4. Type the command `cat /etc/group`
The `cat` command shows the contents of the `/etc/group` file. Notice that the group named `mygroup2` with a groupID of 279 is gone.

Lesson 7.8 Using the Admintool to Create a User Account

In this lesson the Admintool is used to create a generic user account. The Admintool can only be run by the root user, a user within the sysadmin group 14, or an RBAC (Role Based Access Control) user that has permission to use it.

1. Log in as the root user.
2. Open a Terminal window
To open a Terminal window, right click anywhere in unoccupied desktop space, left click on Tools, then left click on the Terminal icon.
3. Type the command `Admintool &`

Special Warning – Do not close down the Terminal or console window in which the `Admintool &` command was typed! If you do, the Admintool will also close down.

Figure 7.5 shows the Admintool display.

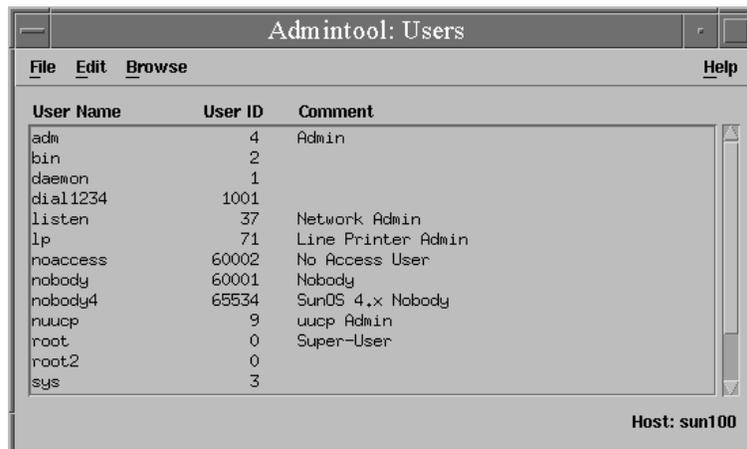


Figure 7.5 The Admintool Display

4. Select **E**dit from the toolbar menu.
5. Select **A**dd from the **E**dit menu.
6. Create a new user with the following settings:
 - Username: user32
 - User ID: Let the system pick this one

- Primary Group: Let the system pick this one
- Secondary Group: Leave this blank
- Comment: User 32
- Login Shell: Bourne
- Password: Normal Password...
*This will require a password. Set the password to **user32**.*
- Min Change: 14
- Max Change: 21
- Max Inactive: 14
- Expiration Date: 20 May 2003
Set the expiration date so that it is after the current system date.
- Warning: 7
- Create Home Directory: Check this box
- Path: /usr/user32

The final screen should look like Figure 7.6

Figure 7.6 Add User Window

7. Click on the **OK** button

You have successfully created a user Account, using the Admintool.

Using the Solaris Management Console (SMC)

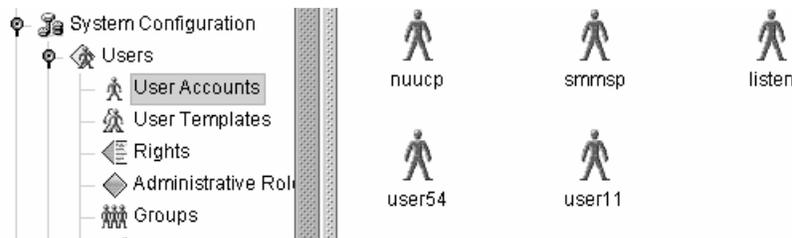
The SMC (Solaris Management Console) includes a GUI tool called the Solaris User Manager 1.4. This tool can be used to create, modify and remove user accounts. The chief advantage of the SMC is that it has several templates and wizards that can be used to create and work with multiple accounts at the same time. This is a nice feature to have when a company needs to bring in two hundred contract workers and two hundred user accounts need to be created for the project.

Lesson 7.9 Solaris User Manager 1.4

In this lesson a user will be created by the SMC wizard.

1. Log in as the root user.
2. Open a Terminal window
To open a Terminal window, right click anywhere in unoccupied desktop space, left click on Tools, then left click on the Terminal icon.
3. Type the command **smc &**
4. Left click on the This computer (sun100) icon.
5. Left click on the System Configuration icon.
6. Left click on the Users icon.

The SMC user manager starts up. The screen will be look like the following:



User Accounts

This is used to add a user or users, to copy selected users to mailing lists and groups, to assign rights to a user, to set up user policies, and to show the properties that are associated with users.

User Templates

These are used to create a template for the creation and management of users. The template includes: settings such as the user's login shell, the user's account availability (permanent or expires after a certain time), the primary and secondary groups the user belongs to, the user's home directory and where the user initialization files come from, the permissions on the user's home directory, and the user's password aging. The user's mail server can be specified, as well as the user's Projects.

Rights

The rights associated with RBAC (Role Based Access Control). The administrator can create new rights or can use pre-defined rights.

Administrative Roles

The root user can create and assign administrative roles to ordinary users. These roles lighten the burden placed on the system administrator.

Groups

This is a simple management GUI for creating groups and assigning users to them.

Mailing Lists

This only applies to a server that is using a mail program such as the **sendmail** program. This icon lets the system administrator configure mailing lists for the users. A mailing list is a collection of the users that have email accounts on an email system.

It's time to go into detail about how to use the User Accounts icon. We will use it to add users, and to work with user policies.

7. Left click on System Configuration
8. Left click on Users and then left click on User Accounts.

A popup window may appear, asking for the root user's name and password. If so, enter them.

*The User Accounts display initially shows the user accounts that are in a system. The accounts with a UID (User ID) of less than 100 are used for internal Solaris 9 usage. For example, the **lp** user is used to print documents. Nobody could actually login using the **lp** account. The **nobody**, **noaccess** and **nobody4** accounts are created for security reasons. Some programs like anonymous FTP and other connectivity programs use the **nobody** and **noaccess** user accounts.*

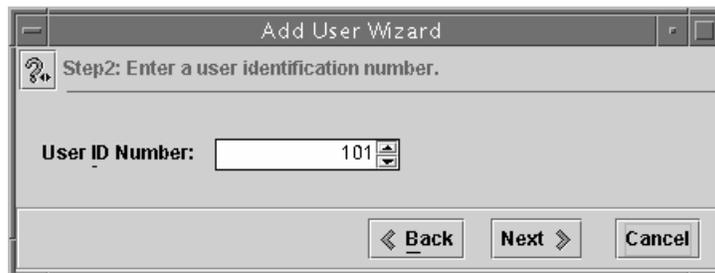
9. Left click on the Action Menu item.
10. Left click on the Add User menu item.
11. Left click on With Wizard...

From here, adding a user is a rather easy process of following the wizard. As shown below, the first screen asks for the login name of the user. Solaris 9 supports 2 to 32 letters, numbers, underscores (_), hyphens (-) and periods (.). The username must begin with a letter and must have one lowercase letter. The Full Name and Description boxes are used to describe a user.



The screenshot shows a window titled "Add User Wizard" with a subtitle "Step 1: Enter a user name.". It features three text input fields: "User Name" containing "user34", "Full Name" containing "user34", and "Description" containing "user34". At the bottom right, there are three buttons: "Back" (disabled), "Next" (active), and "Cancel".

The next screen allows the system administrator to pick a UID between 100 and 2147483647. Remember that a UID less than 100 is used internally by the system. Do not pick an unused number less than 100. The results could be very unpredictable.



The screenshot shows a window titled "Add User Wizard" with a subtitle "Step 2: Enter a user identification number.". It features a single spinner box for "User ID Number" with the value "101". At the bottom right, there are three buttons: "Back" (disabled), "Next" (active), and "Cancel".

A password can be set in the next screen, or it can remain blank until the user types in a new password. For security reasons it is a good idea to set a password at this time. The password has to be between 6 to 15

characters in length. This is longer than the eight character limit imposed by Solaris 8. The password must have two alphabetical characters and one numeric or special character in the first six characters. This is to prevent dictionary words like “potato” or “volcano” from being used. Hackers typically take a dictionary trolling program and try every word in the dictionary as a password. A password like my32passwd would be almost impossible to guess.



The next screen asks the system administrator to pick a primary group. Groups are used for permissions on files and directories. For example, an Engineering group could be used to set read only permission to a directory for everyone in the Engineering group. The only groups that are available are groups that are already in the `/etc/group` file.



The next screen asks for the user's home directory. The default location is `/export/home`. Many systems use `/usr/<username>` as a default home directory. The home directory stores the user's initialization files. These files are from the `/etc/skel` directory. These files are `.profile` for the Bourne and Korn shells, and `.cshrc` and `.login` for the C shell. If a user logs into the CDE, the `.dtprofile` file and `.dt` directory are created in the user's home directory. The home directory can be thought of as the user's “piece of real-estate” on the server. The user can create files and directories in the home directory.



The next screen just shows the mail server and mailbox associated with the user; no configuration is possible in this screen. The Mail server is the host system and the default mail location is `/var/mail/<username>`. The mail system only works if the `sendmail` program has been installed on the workstation or server. Because this screen is so basic, no screen shots were taken.

The final screen reviews all the choices that have been made so far. At this point, the administrator can left click on **Finish** and create the user. After a short delay, the user is created and added to the User Accounts view pane. If the user's account does not show up, left click on View in the toolbar and then left click on Refresh.

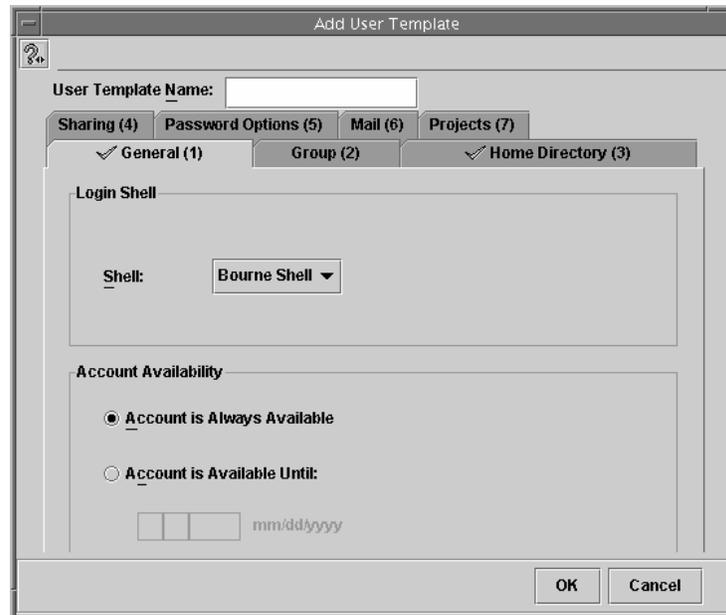


Creating Users from Templates

Templates can be created to add single users or multiple users. Templates allow the system administrator to specify certain common characteristics for a user or group of users. For example, the system administrator could specify that all users have the home directory `/usr/<username>` and that all users work with the Bourne shell (`/bin/sh`).

To add user templates, left click on the User Templates icon. Left click on Action (on the Menu bar) and then left click on Add User Template.

On the top of the screen are text boxes for the template's name and description. The template name can be a maximum of 32 characters, numbers and spaces. The description can be a maximum of 256 letters, numbers and spaces.



There are seven tabs used to create the template:

General Tab

Login Shell

Specify the login shell for the template. These are: Bourne Shell, C Shell, Korn Shell, BASH, T Shell, Z Shell, other (specify a shell).

Account Availability

Account is always available or The Account is Available Until: mm/dd/yyyy.

Group Tab

Primary Group The primary group is used for permissions on files and directories.

Alternate Groups

A user can have an additional fifteen alternate groups. These function just like the Primary Group. All these groups must exist in the `/etc/group` file.

Home Directory Tab

Home Directory Server

Specify the name of the server that holds the user's accounts. This can be the local host or a remote host in the case of diskless clients.

Home Directory Path

The location of the user's home directory. There is a checkbox that is used to specify if the user's login name should be appended to the Path above. For example, if a user named **jadams** has a home

directory path of `/export/home`, his home directory would be created in the `/export/home/jadams` directory.

Copy Initial Files From

These files are `login.profile` (for the Bourne shell)
`login.login` (for the Korn shell)
`login.cshrc` (for the C shell)

These files are copied to `.profile`, `.login` or `.cshrc` file, depending on the shell selected.

Automatically Mount Home Directory

This option lets the user keep the same home directory, even if it is moved to a different location or server.

Sharing

Group Members

Shares the user's home directory read or full permissions with any user that is a member of the same group. (To see what groups a user is a member of, log on as that user and type the command `cat /etc/group`). Users that belong to the same group as this user can have read or full permission to this user's home directory. However, note that giving full permission to anyone in another user's home directory is a very bad idea. That would destroy security and system accountability.

Other Users

This checkbox lets you give other users read permission or full permission to this user's home directory. That is extremely bad for system security. As before, it's unthinkable to give anyone full permission to a user's home directory!

Password Options

User Must Keep for:

Number of days a password must be used before it can be changed. Prevents too many password changes in a short period of time. Some users quickly change their password to a temporary password and then change their password back to the original password. The minimum time stops this type of end-run. The maximum time that a password can stay locked down is 730 days.

Before Change, Alter User

This is a notification message that is sent to the user, letting him or her know that their password will expire soon.

User Must Change Within:

Number of days after which the password must be changed. This prevents a password from remaining forever. After the password is changed, the user must keep that password for the period specified in the "User Must Keep For:" box.

Expires if Not Used For:

This option prevents a forgotten account from remaining active forever. If an account is not used in a certain amount of time, the `/etc/shadow` file will have the `*LK*` flag placed in the password, locking down the account.

Mail

Specify the mail server - The name of the mail server for this template.

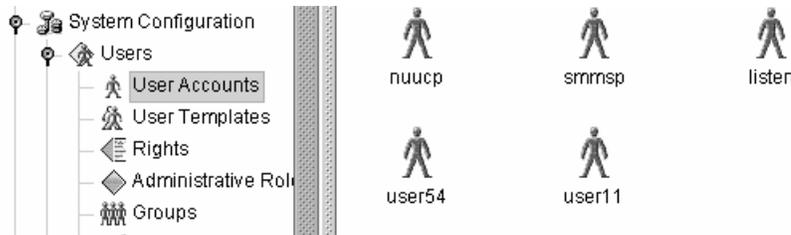
Projects

Specify the user's primary project and secondary projects. A project can be thought of as a collection of programs. This dialog box specifies the projects the user is assigned to.

Lesson 7.10 Create Users from a Template

The SMC has a wizard that allows the system administrator to create multiple users at the same time. This is a nice feature to have when a group of contractors or new employees shows up and the system administrator needs to create two hundred new users.

1. Log in as the root user.
2. Open the Workspace menu.
To open the Workspace menu, right click anywhere in unoccupied desktop space.
3. Left click on the **Tools** menu item.
4. Left click on the **SMC** icon.
5. Left click on the computer (sun100) icon.
6. Left click on the System Configuration icon.
7. Left click on the Users icon.
8. Left click on the User Accounts icon.
9. Left click on the **A**ction menu item.
10. Left click on Add **M**ultiple Users.



This

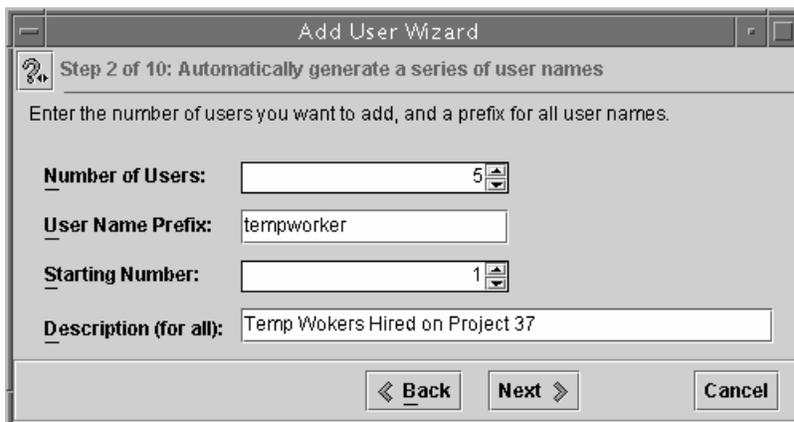
*This wizard is very similar to the first add user wizard, except that it is used to add multiple users to the system. Select **With Wizard**. The first screen how to add new usernames:*

- Specify a text file
- Type each name
- Automatically generate a series of usernames ← *select this option*



The **Automatically generate a series of usernames** method is really neat. Just type in a prefix name (such as *tempworker*) and a starting number (such as *1*) and the wizard automatically adds extra users, each with a unique prefix name-number.

11. In this example with a prefix name of *tempworker* and a starting number *1*, five users could be created: *tempworker1*, *tempworker2*, *tempworker3*, *tempworker4* and *tempworker5*.



12. The next couple of screens look just like the ones in the Add User template, except for a couple of key differences. The login shell can be chosen for the temp workers. The administrator can choose from the Bourne Shell, the C shell, the Korn shell, BASH, T Shell and the Z shell. There is also an "other" option that lets the administrator select a shell. Most shells are located in the */bin* directory. Type the command `cat /etc/passwd` to view the shells that are set up for the users. The Bourne shell is */bin/sh*, the C shell is */bin/csh* and the Korn shell is */bin/ksh*. This screen also lets the administrator choose if the account is always available or locked after a certain date.



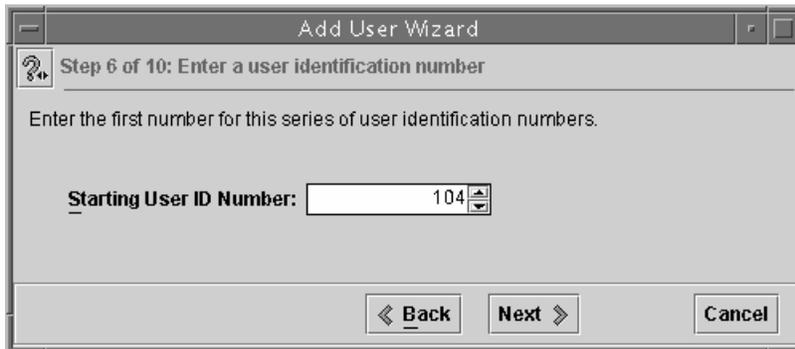
13. The next screen looks just like step 3 in the Add User wizard. The password setting can be set to "Users Must Set Password At First Login" or "Users Must Use This Password At First Login:" which lets the administrator set the user's password.



Unlike in the Add User wizard, the Multiple Add User wizard lets the administrator set password aging. This information is stored in the `/etc/shadow` file.



The next step lets the administrator select the starting UID for the series of users. Remember that the UID is defined in the `/etc/passwd` file. The UID is a numeric representation of the user that the Solaris 9 operating system likes to use. Enter the starting number in the text box.



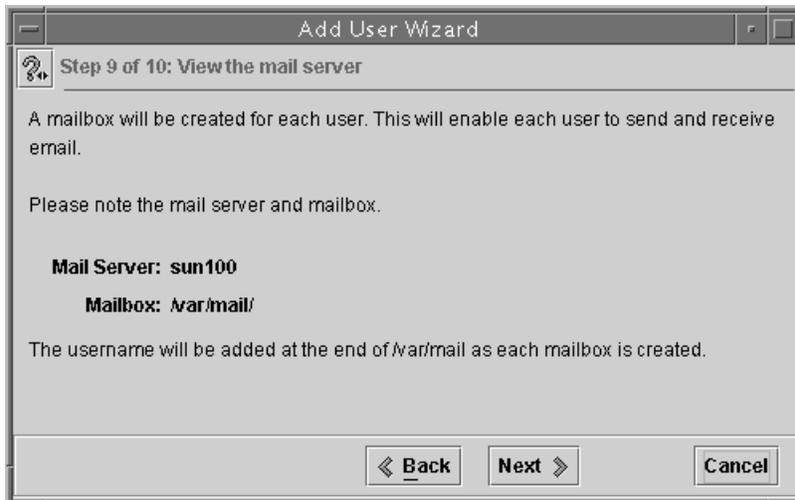
14. The next screen gives a drop down menu choice for the primary group. The primary group is used for permissions on files and directories. The **Staff** group is the default group for a user. Choose the group for the new users.



15. This screen lets the administrator choose the home directory for the group of users. Enter the directory in the text box.



16. If you check **Automatically Mount Home Directory** the user's home directory is mounted, even if it is moved to another server.
17. The next screen shows the mail server and mailbox for the users. The mail server is the local host and the Mailbox is : `/var/mail/<username>`.



18. The final screen is just a confirmation screen. If anything needs changing, click < **Back** to reach that information, then proceed until this screen appears again.

When all information is correct, click **Finish**. It may take quite some time to create all the user accounts specified in the Add Multiple Users wizard. A popup window appears, stating that the users will be added in the background.

Click on **View, refresh** to see the users added to the system. Check the Log View to check if there are any problems. This process is very slow; adding, just 5 users on a SunBlade 100 took about two minutes to complete.



How Name Services Work with Users and Groups

A Name Service can be used to centralize administration of user accounts by providing the three critical files `/etc/passwd`, `/etc/group` and `/etc/shadow` from a centralized server. When someone tries to log in from a workstation, the workstation contacts either a NIS, NIS+ or LDAP server. These servers have all of the user's

information in their records. A system administrator does not have to choose only one source. A workstation could theoretically check for a user's name or a group's name both from a NIS+ server and LDAP server.

Most networks start off just using the `/etc/passwd` file to look up users on each server. But when the number of servers becomes large, a NIS server might be installed. As the network continues to grow, the NIS server may be converted into a NIS+ server. The NIS+ server has more functionality and power than a NIS server. If the network continues to grow further, the NIS+ server may be converted into an LDAP server. This is not a hard and fast rule. For example, some companies could have a huge network of UNIX boxes but never use NIS+. Name services like **NIS (and NIS+)**, **DNS** and **LDAP** are covered in Chapter 23, Chapter 29, and Chapter 30 respectively.

Quota Limits

After the user's account is created, a quota can be set up on the user's account. This quota will limit the amount of disk space that a user is allowed to have. This is necessary because the average Solaris server puts the user's home directories all on the same file system. If one user uses up too much space in the `/export/home` directory, the slice the `/export/home` directory resides on can become full. If this happens, none of the users can read or write files, creating a big problem for a system administrator.

Terms Used with Quotas

Blocks The UFS file system uses disk blocks for storage. The default block size in Solaris is 1024 bytes or 1 KB. Quotas control the number of data blocks that a user can use when creating files.

Inode An inode stores all of a file's information, such as date created, file owner, size of the file, and data blocks associated with the user. Each file has an inode assigned to it. The inodes are stored in the inode table of the cylinder group. The UFS file system has a limited supply of inodes.

Soft Limit A soft limit is flexible in that the user can create more files, or larger files, than the intended limit. Once the user exceeds the soft limit, he or she has a set amount of time to reduce the size or number of files. After a given amount of time, the server will prevent the user from creating more files or will prevent the user from enlarging current files.

Hard Limit Most systems use a hard limit, as well as a soft limit. The hard limit is an absolute limit to the size and number of files created.

Setting Quota Limits

Quotas can be set for two parameters: the number of **inodes** that a user can use (an inode stores information about a single file), and the number of **blocks** that a user can use (1 block = 1/2 KB). There are two types of limits, a soft limit and a hard limit. These limits can only be changed with the quota command for each user. To disable quotas for a user, set the hard and soft block and inode limits to zero, and disk quotas for that particular user will be disabled.

If one user created a 10 GB file, that file could easily fill up the file system dedicated to users' home directories. With a quota, a user can not use too much disk space. Also, there are only so many files that a file system can contain, because the file system has a limited number of inodes. The quota can also be set up to limit the number of inodes dedicated to a user, and hence, the number of files that the user can create.

When setting up the quota limitation, understand that the CDE creates a directory named `.dt` in the user's home directory. This directory uses about 52 KB of disk space. The Netscape Navigator creates a directory named `.Netscape` (also in the user's home directory) that takes up about 314 KB. The typical "housekeeping" directories

for CDE, Netscape, and other programs contain about 90-150 files. The disk space used in “housekeeping” is about 25 MB.

User quotas are essential for an ISP (Internet Service Provider) that uses Sun Solaris for its customers' Internet accounts. One way to set up quotas is to give each user an equal share of the file system's resources. Another way is to give each user a slightly higher quota than his or her fair share. The theory is that not all users are going to use their capacity, so each user can have a slightly larger quota, just in case. However, if each user is given this type of slightly larger quota, carefully monitor the file system, to avoid running out of disk resources.

Make sure not to set a quota limit that is so low that the user will not be able to use the CDE or Netscape without problems. Most quota limits are rather high, like 100 MB, so this is not an issue. For servers with exceptionally small hard drives (< 2 GB in size) and with very small quotas, this could be an issue.

Commands Used with Quotas

edquota	Creates the quota limits for a user
quoton	Turns on user quotas
quotacheck	Shows the overall file system's quota usage.
quota	Shows an individual user's quota.
repquota	Used to show the quota on a file system.

Lesson 7.11 Setting Up a Quota

In this lesson the reader will create a test user named user53 (if user53 does not already exist). The user53 account will be activated by logging in, and then the user account will start the Netscape browser. After user53 logs out, the root user will log back in again. After the user's current resources are calculated, the `quota` command will be used to set up a quota, with soft limits of 10 MB of files and 1000 inodes, and with hard limits of 20 MB of files and 2000 inodes.

1. Log in as the root user.
2. Open a Terminal window.
To open a Terminal window, right click anywhere in unoccupied desktop space, left click on Tools, then left click on the Terminal icon.
3. Type the command **useradd -m -d /usr/user53 user53**
*If user53 exists, create another user with the format **user<##>** like **user32**, **user48** or **user86**.*
4. Type the command **cp /usr/user11/local.profile /usr/user11/.profile**
5. Type the command **passwd user53**
*The **passwd** command creates a new password for user53. If a different username was chosen in step 3, use that username and password instead. The message on the screen should read:*
New password: < the password should be **user53** >
Re-enter new password: < re-type the password **user53** >
*If for some reason the **passwd** command can not be used to change the user's password, type the command **admintool** & and use the Admintool to change the user's password.*
6. Exit the CDE.
7. Log in as user53 (or whichever username you have created). Continue to use that name anywhere where it says "user53" below.
8. Open Netscape Navigator.
The Netscape Navigator icon is the on the Front Panel window on the far left side.



It looks like a globe with a red clock hand over the globe. The Netscape Navigator will be opened and closed to create some files that user53 owns.

9. Exit the CDE.
10. Log in as the root user.
11. Open a Terminal window
To open a Terminal window, right click anywhere in unoccupied desktop space, left click on Tools, then left click on the Terminal window icon.
12. Type the command `du -h /usr/user53`
The command `du -h /usr/user53` shows a summary of the user's home directory usage in human-readable format. The command `du -k /usr/user53` can also be used to view the contents of user53's directory. The output of the `du -h` command should look something like the following:

```

2K /usr/user53/.dt/sessionlogs
1K /usr/user53/.dt/types/fp_dynamic
2K /usr/user53/.dt/types
1K /usr/user53/.dt/icons
 1K /usr/user53/.dt/appmanager
19K /usr/user53/.dt/help/user53-sun100-0
20K /usr/user53/.dt/help
4K /usr/user53/.dt/sessions/current
9K /usr/user53/.dt/sessions
1K /usr/user53/.dt/tmp
39K /usr/user53/.dt
1K /usr/user53/.java/.userPrefs
2K /usr/user53/.java
53K /usr/user53

```

13. Type the command `find /usr/user53 -name "*" | wc`

This command should display the following output:

```
27 27 855
```

*The `find` command finds all the files and directories in the `/usr/user53` directory because it is searching for any file or directory with any name (`-name *`). The output is then piped into the `wc` (word count) command. This counts the number of files and directories listed by the `find` command.*

The first column of this command shows the total number of files and directories in the user's home directory. Don't set the user's inode limit below this number, or the user will not be able to create any more files. It's a good idea not to set the quota limit to less than 100 files. An upper limit of 5,000 files should be reasonable for an ordinary user.

14. Type the command `cp /etc/vfstab /etc/vfstab.backup`
This step copies the `/etc/vfstab` file to `/etc/vfstab.backup`. If a mistake is made, type the command `cp /etc/vfstab.backup /etc/vfstab` to restore the original file.
15. Edit the `/etc/vfstab` file so that the file system that holds the user's home directory will have the mount option `rq` instead of the dash (-) symbol. If the `/usr` file system is not on a separate slice, it is created under the root slice (/).

The original `/etc/vfstab` file looks like this:

#device	device	mount	FS	fsck	mount	mount
#to mount	to fsck	point	type	pass	at boot	options
fd	-	/dev/fd	fd	-	no	-
/proc	-	/proc	proc	-	no	-
/dev/dsk/c0t0d0s1	-	-	swap	-	no	-
/dev/dsk/c0t0d0s0	/dev/rdisk/c0t0d0s0 /	/	ufs	1	no	-

The modified `/etc/vfstab` file looks like this:

#device #to mount	device to fsck	mount point	FS type	fsck pass	mount at boot	mount options
<code>fd</code>	-	<code>/dev/fd</code>	<code>fd</code>	-	<code>no</code>	-
<code>/proc</code>	-	<code>/proc</code>	<code>proc</code>	-	<code>no</code>	-
<code>/dev/dsk/c0t0d0s1</code>	-	-	<code>swap</code>	-	<code>no</code>	-
<code>/dev/dsk/c0t0d0s0</code>	<code>/dev/rdsk/c0t0d0s0</code>	<code>/</code>	<code>ufs</code>	<code>1</code>	<code>no</code>	<code>rq</code>

Notice at the lower right of each display that the dash (-) symbol has been replaced with the keyword `rq` for the root (/) file system.

16. Type the command `touch /quotas`
The `touch` command creates an empty file named `quotas`. This is necessary to set up disk quotas.
17. Type the command `chmod 600 /quotas`
The command `chmod 600 /quotas` sets the file permissions to read/write for the root user only. This is necessary so a hacker can not modify this file and break the file system.
18. Type the command `/usr/sbin/quotaoff /`
19. Type the command `/usr/sbin/quotaon /`
20. Type the command `/usr/sbin/edquota user53`
The last three commands ensure that the quota is changed correctly.
21. When `edquota` runs, it starts up a simple text editor. You should see the line
`fs / blocks (soft = 0, hard = 0) inodes (soft = 0, hard = 0)`
22. Change the user's soft and hard limits to the following:
`fs / blocks (soft = 10240, hard = 20480) inodes (soft = 1000, hard = 2000)`
Remember that **1024 KB = 1 MB**
1 block = 1/2 KB
2048 blocks = 1 MB
Save the file by left clicking on the `File` menu bar item, then `Save`.
Left click on `File`, then left click on `Close`.
23. Type the command `/usr/sbin/quotaon -va /`
The `quotaon` command starts quota checking on the root file system.
The following message should appear:
`/ quotas turned on`
24. Type the command `/usr/sbin/quotacheck -fvp /dev/dsk/c0t0d0s0`
The `quotacheck` command shows the quota on the root file system.
A message like the following should appear:
`/dev/rdsk/c0t0d0s0: user53 fixed: files 0 -> 103 blocks 0 -> 9228`
25. Type the command `df -k`
The `df -k` command shows the used and free disk space on all file systems (in kilobytes). The output should be something like: `6279721 available kilobytes`. The command `df -h` (the `-h` option is only available in Solaris 9) can also be used to show the output in human-readable format, such as `3.0G` for 3.0 gigabytes.
26. Exit the CDE.
27. Log in as user53.
28. Open a new Terminal window.
To open a Terminal window, right click anywhere in unoccupied desktop space, left click on `Tools`, then left click on the Terminal icon.
29. Type the command `/usr/sbin/mkfile 100m 100mbfile`
The command `mkfile` tries to make a 100 megabyte file, which is past the quota limit. A message like the following should appear:
`quota_ufs: Warning: over disk limit pid 1010, uid 110, inum 82970, fs /)`
`quota_ufs: over hard disk limit (pid 10101, uid 110, inum 82970 fs /)`
`500mbfile: initialized 21413888 of 104857600 bytes: Disc quota exceeded`

30. Type the command `ls -lh 100mbfile`.
The output shows that although the file 100mbfile was created, it has a one byte size. This is a strange flaw in the quota file system in that when a user's quota is exceeded, it lets the user create a file but only puts one byte in it.
31. Type the command `du -ah /usr/user53`
This shows how much space is taken up by all files and directories in the user's home directory.
32. Now type the command `df -k`
The `df -k` command displays the available space on all file systems. Notice that there are 20 MBs less disk space than there were before. This is due to a flaw in the quota file system. The quota file system will tell the user that he or she has exceeded the quota limits, but will still create a 0 or 1 byte file with the desired file name. The user's maximum quota disk space is then used up, even though the system never created the large file that the user requested.

In this case, when the user tried to create a 100 MB file, the error message appeared, saying that the quota had been exceeded. As mentioned above, 20 MBs of disk space was then used up for user53, even though only a 0 or 1 byte file was created.

Checking Quotas with repquota

The system administrator can check the quotas on a file system. Figure 7.7 is a screen shot of the output from the `repquota` command.

User	used	soft	hard	timeleft	used	soft	hard	timeleft
user53	+-	23655	10240	20480	NOT STARTED	110	1000	2000
user11	+-	20952	10240	20480	7.0 days	2	1000	2000

Figure 7. 6 Output from the `repquota` Command

Lesson 7.12 The `repquota` Command

In this lesson the `repquota` command is used to show the quotas used by different users on the root file system. In this lesson `user54` is created for future lessons. This lesson also illustrates that there is sometimes a time lag between when a quota is enabled for a user and when the file system recognizes the quota limitations. The commands `quotaoff` then `quotaon` can be used to stop and start quotas on a file system.

1. Log in as the root user.
2. Open a Terminal window.
To open a Terminal window, right click anywhere in unoccupied desktop space, left click on Tools, then left click on the Terminal icon.
3. Type the command `useradd -m -d /usr/use54 user54`
4. Type the command `cp /usr/user54/local.profile /usr/user54/.profile`
5. Type the command `passwd user54`
The message on the screen should read:
New password: < the password should be user54 >
Re-enter new password: < re-type the password user54 >
6. Type the command `edquota user54`
7. Give the new user a 10 MB lower limit, a 20 MB upper limit , a 1000 soft inode limit and a 2000 hard inode limit:
fs / blocks (soft = 10240, hard = 20480) inodes (soft = 1000, hard = 2000)
8. Type the command `repquota -v /`
A message like the following should appear:
/dev/dsk/c0t0d0s0 (/):

Block limits

File limits

User		used	soft	hard	timeleft	used	soft	hard
user53	+-	20469	10240	20480	6.9 days	49	1000	2000

As can be seen in the above example, *user54's* quota is not yet running, because there is no listing for that user.

- Type the command `quotaoff /`
- Type the command `quotaon /`
- Type the command `repquota -v /`

Now *user54* should be seen in the message:

```
# repquota -v /
/dev/dsk/c0t0d0s0 (/):
```

Block limits				File limits				
User		used	soft	hard	timeleft	used	soft	hard
user53	+-	20469	10240	20480	6.9 days	49	1000	2000
user54	--	0	10240	20480		0	1000	2000

As shown above, when a new user is added to the system, turn off the file system's quota, then turn it back on, to get the user's quota working.

- Type the command `quota -v user54`

The screen should display the message

```
Disk quotas for user54 (uid 394):
Filesystem  usage quota limit   timeleft files quota limit   timeleft
/           0  10240 20480    0         1000 2000
```

Changing a User's Quota Limits

After a user's quota has been created, it can be changed. Quotas can also be removed from a user by setting all the block and inode hard and soft limits to zero (0). This is recognized by the quota system as a disabled quota file system. It is also possible to disable only part of a user's quota. For example, if a server does not need a hard and soft limit on inodes, the hard and soft limits for the inodes can be set to zero (0) while the hard and soft block limits can still remain intact. It is also possible to limit the number of inodes created, but to disable hard and soft disk block quota usage, by setting the block quotas to zero.

Lesson 7.13 Changing a User's Quota Limit

In this lesson *user54's* quota limits will be disabled by changing all the hard and soft limits to zero (0). The `quota -v` command is then used to display *user54's* quota limits (none). Next, *user54's* quota limits will be reinstated by editing them with the command `edquota`. Then, the command `quota -v` will be used again to see the revised quota limits.

- Log in as the root user.
- Open a Terminal window.
To open a Terminal window, right click anywhere in unoccupied desktop space, left click on Tools, then left click on the Terminal icon.
- Type the command `edquota user54`
- To disable *user54's* quota limit, change the hard and soft limits for blocks and inodes to zero (0).
fs / blocks (soft = 0, hard = 0) inodes (soft = 0, hard = 0)
- Make sure that the quota has been changed for *user54* with the command `quota -v user54`
The following message should be seen:
Disk quotas for user54 (uid 394):
Filesystem usage quota limit timeleft files quota limit timeleft
As shown here, user54's quotas are now turned off. There should not be anything displayed for this user, other than the column headings.
- Type the command `edquota user54`

- Put the previous limits back again with the command **edquota user54: fs / blocks (soft = 10240, hard = 20480) inodes (soft = 1000, hard = 2000)**
- Check user54's limits again with the command **quota -v user54**
User54 should have the original file system limits again.

Disabling a File System's Quota Limits

Quotas can be disabled from a file system by using the **quotaoff** command. When the file system's quota is turned off, users have no limits on file size or on number of files (inode limits). When the file system's quota is turned off, a user's quota settings are not destroyed; they are just disabled. When the command **quotaon** is used to turn on quotas again, the user's quota settings are restored to their previous values.

If the file system quota is turned back on, the users will be accountable for the quotas again. One problem that can occur is that if a user exceeded the allowable file space when quotas were turned off, that user will not be able to make any new files when quotas come back into effect.

If quotas have been turned off for awhile, always alert all users when quotas are restarted. Otherwise, angry users will call, asking why they can not create any files. A good way to see who might be affected is to use the **du -ks /usr/<user_name>** command to view each user's current disk usage. Send an email message to all users who have exceeded the newly re-established limits. Warn them of the upcoming quota changes, and tell them that they should delete any large size files.

Lesson 7.14 Disabling a File System's Quota Limit

In this exercise the quotas on the root (/) file system will be disabled. After the quotas have been disabled, user54 will create a 100 MB file (if your hard drive has this space). Next, quotas will be turned back on again. User54 will not be able to create any more files until the 100 MB file is deleted from the root file system.

- Log in as the root user.
- Open a Terminal window.
To open a Terminal window, right click anywhere in unoccupied desktop space, left click on Tools, then left click on the Terminal icon.
- Type the command **quotaoff -v /**
*The command **quotaoff -v /** disables file system quotas for the root (/) file system. The following message should appear:*
/: quotas turned off
- Type the command **rm /quotas**
*The **rm** command is used here to delete the quotas file.*
- Edit the **/etc/vfstab** file and change the line that reads:

/dev/dsk/c0t0d0s0	/dev/rdsk/c0t0d0s0	/	/	ufs	1	no	rq
--------------------------	---------------------------	----------	----------	------------	----------	-----------	-----------

to its original settings. The line should look something like this:

/dev/dsk/c0t0d0s0	/dev/rdsk/c0t0d0s0	/	/	ufs	1	no	-
--------------------------	---------------------------	----------	----------	------------	----------	-----------	----------
- Change to user54 with the command **su - user54**
The password for user54 should be user54
- Type the command **/usr/sbin/mkfile 100M 100mbfile**
This creates a 100 MB file. If quotas were turned on, this last step would not be possible.

Key Points to Remember

It is important that a Solaris system administrator know how to add users to the system, using both the GUI tools and the command line utilities. The GUI tools let the system administrator *see* the users that are being created. The GUI tools are also very nice when it comes to creating a lot of users from a template.

GUI tools are nice, but what does a system administrator do when he or she has a server without a GUI interface? Most companies do not have a monitor, keyboard and mouse attached to the server. The commands **useradd**, **usermod** and **userdel** create user accounts when there is no GUI capability available.

It is OK to *play around* with ordinary user accounts. But be very careful when working with the root user's account. If the root user's account becomes damaged, a server might not even be able to start. Understand the Solaris 9 operating system uses the root user's account internally and that if the account is disabled or damaged, some key process might not be able to start.

Chapter 8 Working with Directories and Files

Lessons in This Chapter

Lesson 8.1 Understanding the <code>/dev</code> and <code>/devices</code> Directories.....	8-2
Lesson 8.2 Understanding the <code>/bin</code> Directory.....	8-3
Lesson 8.3 Creating Soft Links to a Directory	8-7
Lesson 8.4 Soft and Hard Links to a File	8-8
Lesson 8.5 Using the <code>ls</code> Command.....	8-10

Introduction

There are certain key directories that are used in Solaris 9. It is a good idea for a system administrator to know about them. If a configuration file is bad, knowing that the `/etc` directory holds the system's configuration directories would be a starting point for troubleshooting Solaris 9. This chapter covers some of the more common system directories in Solaris 9.

The `/dev` and `/devices` Directories

The `/dev` directory is used to hold all the device files. Every physical device attached to a server is represented by a file in the `/dev` and `/devices` directory. If a modem is attached to a server, there is a file located under the `/dev` directory and `/devices` directory that represents the modem. The Kernel reads and writes to files in the `/dev` and `/devices` directory to operate the modem. The modem's device driver (software that interacts between a device and the operating system) watches the files in the `/dev` and `/devices` directories and manipulates the modem according to what the operating system requests. Figure 8.1 shows how a device is operated in Sun Solaris.

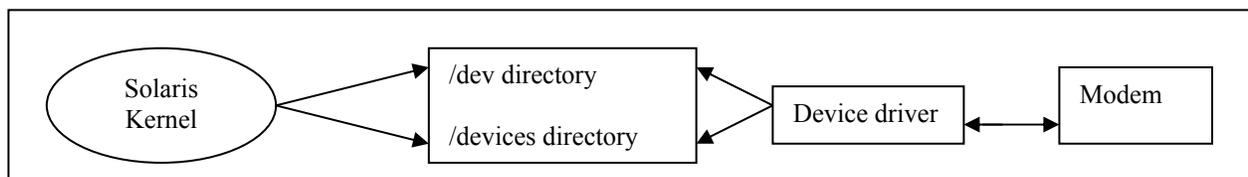


Figure 8.1 Illustration of a Device Driver

The `/dev` directory has subdirectories that reference certain devices. Every file in the `/dev` directory is a symbolic link to a matching file in the `/devices` directory. The file names under the `/dev` directory are known as "symbolic device files." These file names are easier to remember and work with than the files in the `/devices` directory. For example: `/dev/dsk/c0t0d0s0`. The files under the `/devices` directory are called "device files." These files are very hard to remember and type. For example, in the `/devices` directory on the Author's Sun Blade 100 there is the device:

```
/devices/pci@1f,0/ide@d/dad@0,0:a
```

The same device is represented by the file

`/dev/dsk/c0t0d0s0`

That is a much smaller and easier file to work with.

In a nutshell, the `/dev` directory was created for humans, so it uses file names that are easy to understand. The `/devices` directory was created for the Kernel and has very complex file names that give a great deal of technical information that the Kernel needs.

Lesson 8.1 Understanding the `/dev` and `/devices` Directories

In this lesson the reader will explore the different files located in the `/dev` and `/devices` directories, and how these files relate to the hardware attached to the workstation.

1. Log in as the root user.
2. Open a Terminal window
To open a terminal window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Tools menu item, then left click on the Terminal window icon. The next three lines produce a listing of the `/dev/dsk/c0t0d0s0` file.
3. Type the command `cd /dev`
4. Type the command `cd dsk`
5. Type the command `ls -l c0t0d0s0`
The output of this command should be something like

```
lrwxrwxrwx1    root    root    38    Sep 30  2002    c0t0d0s0  
->..././devices/pci@1f,0/ide@d/dad@0,0:a
```

The file `/dev/dsk/c0t0d0s0` is a symbolic file name that points to `/devices/pci@1f,0/ide@d/dad@0,0:a` device file.
6. Use the `ls` and `cd` command to explore some other device directories under `/dev`

```
/dev/cus    – Used for modems and dial out devices  
/dev/dsk   – Hard drives or access devices by blocks  
/dev/fbs   – Frame buffer files  
/dev/fd    – File descriptors /dev/fd/0 – stdout, /dev/fd/1 –stdin,  
              /dev/fd/2 – stderr  
/dev/pts   – Pseudo terminal  
/dev/rmt   – Magnetic tapes  
/dev/term  – Terminal connections  
/dev/rdsk  – Raw device interface (access a device in character mode)
```

7. Type the command `ls -l /devices`
The `/devices` directory – Contains references to devices by their physical names.

The `/bin` Directory

Most user commands are in the `/bin` directory. The `/bin` directory is only a symbolic link to the `/user/bin` directory.

Lesson 8.2 Understanding the `/bin` Directory

In this lesson the reader will change to the `/bin` directory. The reader will take note that the current directory is now the `/usr/bin` directory. This lesson emphasizes the point that the `/bin` directory is only a symbolic link to the `/usr/bin` directory. The `which` command is then used to find the location of the `ls` command.

1. Log in as the root user.
2. Open a Terminal window
To open a Terminal window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Tools menu item, then left click on the Terminal window icon.
3. Type the command `cd /bin`
4. Type the command `pwd`
As can be seen in this example, the `/bin` directory points to the `/usr/bin` directory.
5. Type the command `ls`
Recognize any commands here?
6. Type the command `which ls`
The `which` utility shows the location of a command.
7. Type the command `which <any_other_command>`

The `/etc` Directory

The `/etc` directory holds almost all the configuration files associated with Solaris 9. There are some configuration files located in other areas, but they are an exception.

<code>/etc/acct</code>	This directory holds the system accounting information.
<code>/etc/cron.d</code>	This directory contains the allow and deny files associated with the <code>cron</code> command and the <code>at</code> command.
<code>/etc/default</code>	This directory has the <code>login</code> , <code>su</code> , and <code>passwd</code> files to control user and root access. It also contains the default configuration files for some system packages.
<code>/etc/inet</code>	This directory is the main configuration directory for network services.
<code>/etc/lib</code>	This directory contains dynamic link libraries.
<code>/etc/lp</code>	This directory has printer configuration information.
<code>/etc/mail</code>	The <code>sendmail</code> program stores its configuration information in this directory.
<code>/etc/nfs</code>	This directory is used when an NFS server is set up on the system.
<code>/etc/openwin</code>	This directory is only used if the Open Windows desktop is being used.
<code>/etc/opt</code>	This directory holds configuration information for optional software packages and third-party software packages.
<code>/etc/rc#.d</code>	This directory contains run control scripts, which control how Sun Solaris starts and finishes.

/etc/skel This directory has the **default.profile**, **default.csrc** and **default.korn** shell initialization files. These files are the templates for the **.profile**, **.csrc** and **.korn** files used in user's home directories.

The /export Directory

This directory contains files and directories that are usually exported. The user's home directory (**/export/home**) is under this directory. If a site has a public FTP site, the home directory for the FTP site would most likely be located under the **/export** directory.

The /home Directory

The **/home** directory holds the user's home directories. A user's home directory can be located under **/home/<username>** or **/export/home/<username>**, depending on the setup of the system. This is only a convention. Technically, a system administrator can set up the home directory anywhere on the server. For example, the **user11** user created earlier has a home directory under the **/usr/user11**. A user's home directory can be thought of as his or her "piece of real-estate" on the server. This is where a user has permission to write files and directories. The user's profile files (files that save information such as the user's preferred font, system variables, and path) are saved in this directory. Any time a user types the command **cd** without any arguments, the user is transported back to his or her home directory.

The /kernel Directory

This directory contains the various components that make up the Kernel and Kernel modules. The Kernel can be thought of as the "main program" that runs Sun Solaris. Kernel modules are used to dynamically build a Kernel that meets the hardware and software requirements of the system. The reason Kernel modules are used is to make the Kernel as small and efficient as possible. Why have device drivers and software code that are unnecessary in the Kernel?

The **/kernel** directory has the following sub-directories:

/kernel/drv	Pseudo-device drivers and loadable device drivers.
/kernel/exec	Modules that are required for some executable files and scripts.
/kernel/fs	Modules for different types of file systems, like the proc, UFS and NFS file systems.
/kernel/misc	Miscellaneous modules that do not really fit in any other directory.
/kernel/sched	Operating system schedulers needed by the Kernel.
/kernel/strmod	System V loadable modules.
/kernel/sys	System calls and system accounting information calls.

The /mnt Directory

This is a conventional mount point for Sun Solaris. This directory is empty by default. Most system administrators will create a subdirectory like **/mnt/ftp-public** under this directory. This directory can be used for things like **mount <2nd/CD-ROM> (/mnt/cdrom2)** where the mount directory is used as the base mount

point for other devices. The important thing to understand is that the **/mnt** directory does not contain critical files. It's just a convention used for mount points.

The /opt Directory

This directory usually holds third-party software and optional software that comes with Solaris 9. For example, if a user loaded the Star Office 5.2, the directory **/opt/office52** would appear in the **/opt** directory. This directory holds most of the software programs that are used in the Star Office 5.2 suite.

The /sbin Directory

This directory holds a collection of system maintenance tools that in most cases can be used only by the root user. This directory is accessible if the server's operating system is damaged and the system administrator needs to perform emergency operations on the server.

The /tmp Directory

Any file saved to this directory is a temporary file, and is stored in memory, rather than on a physical disk. These files have very fast read/write times, but they disappear if the system is rebooted or loses power.

The /usr Directory

This directory contains most user commands. There are several important **/usr** subdirectories that should be recognized for certification exams:

/usr/bin	Most user commands are in this subdirectory.
/usr/ccs	C programs are stored here.
/usr/demo	Demonstration programs are sometimes stored in this directory.
/usr/dt	The CDE (Common Desktop Environment) software and configuration files are stored in this directory.
/usr/include	Holds linked files for C programs.
/usr/java	Different Java versions, linked files, executables, anything else associated with the Java Runtime Environment and the Java Software Development Kit.
/usr/lib	Can contain programming libraries for software developers.
/usr/openwin	OpenWindows software programs and files. Understand that Open Windows is not a supported desktop in Solaris 9, but this directory contains some popular Open Windows desktop applications.
/usr/opt	Configuration files usually associated with third-party software packages.
/usr/pub	Different public files, like man pages and online webpages are stored in this directory.

`/usr/spool`

This directory that contains software patches and software packages. This directory is also used by printers for print jobs.

Types of Files

The only things that reside on a UNIX file system are files. Basically, everything on a Sun Solaris hard drive is a file. A directory is nothing more than a file that contains the names of other files.

There are several distinct types of files on Solaris 9:

Device files	Reference a device (such as a modem or hard drive).
Symbolic links	A symbolic link only points to another file, used to reference files on another file system or hard drive.
Hard links	A file name that points to a particular inode.
Directory	Under UNIX, everything is a file. A directory is a file that has the filenames of other directories and files under its dominion.
Regular Files	These files contain text or binary data.

In the section that follows, we'll examine each type of file in detail.

Understanding Device Files

When a physical device (such as a modem, hard drive, or CDROM) is attached to a server, a device file is created to represent that device. When a user copies a file to a hard drive, he or she is in reality copying the file to a device file that represents the hard drive. The operating system watches the file transfer and copies the file to the hard drive behind the scenes.

A device file has two numbers associated with it, a major number and a minor number:

Major Number – The major number is associated with a device driver. For example, a modem could use device driver #202. To use the modem, the server would read the device file associated with that modem (`/devices/pci@1d,0/modem@202,0`). The operating system would “see” that the major device number is 202 and would use the device driver that is associated with 202 with this modem.

Minor Number – If a server has more than one similar device installed, it references the minor number to select which device to use. In the previous example, if a server was connected to a multi-modem system, it could theoretically have 24 similar modems to choose from. If the system needed to use the thirteenth modem, it would look in the `/devices` directory for the entry `/devices/ci@1d,0/modem@202,13`.

You can see the major and minor device numbers by using the `ls -l` (little L) command within the `/devices` directory.

There are two types of device files associated with almost every device: Character Device files and Block Device files. When information is passed to a *character device* file, that information is copied in units of 512 bytes. Character mode is also known as “raw mode”. When a file is copied to a *block device* file, that information is copied in 8 KB chunks.

The use of block device versus character device files usually depends on the kind of device being accessed. Most hard drives and CDROM drives transfer their data via block files or in 8 KB intervals. Modems and terminals usually use send information via character special files or 512 byte chunks due to their rather limited ability to send

large amounts of data. Basically, a block or character file determines how much data is being sent to a device at a time.

Unlike Microsoft Windows, in which a file just has a one filename and one location on the hard drive, in a UNIX environment a file can have more than one filename, and that filename can exist in different locations. The data in a file, and its filename, are two distinct concepts. In Solaris, a chunk of 20 MB of data can have 10 different file names that all point to the same 20MB of data. This is the same as someone having different names that all refer to him—John Anderson, Johnny, Employee 203, Mr. Anderson, PFC Anderson, my best friend, and “Johnny A” can all refer to the same person.

Understanding Hard Links

Hard links are basically filenames on the same level. Each filename or “hard link” has the same precedence as every other hard link. There is no such thing as “This is the only filename associated with the file.” A user can have a file named `mydata.dat` in his or her home directory. Later, that same user can create a hard link named `gooddata.pov` and `oracledat.dat1` and have them refer to the same chunk of data. To call up or copy that data, any of the three filenames can be used. It's important to note, though, that if all the hard links to the data are deleted, the file itself is deleted.

Understanding Soft Links

A symbolic, or soft, link can be thought of as a *creature* that only points to a real file. If a user copies a file named `mydata.dat` to a directory and then creates several *symbolic* links to that data, the symbolic links only point back to the original file, `mydata.dat`. A symbolic link is only a reference file that points back to the original file. If you delete a symbolic link, you will not damage the original file or its data.

So why does UNIX use both hard and symbolic (soft) links to files? The answer is simple. *Symbolic links* can be deleted without destroying the data in the file. A file only exists if the file has at least one *hard link* in existence.

Another reason to have both hard and soft links is that a hard link cannot refer to a file on another file system. For example, a user cannot create a hard link on one hard drive to a file that exists on a different hard drive. A soft link lets the user create a link from one file system to a file on another file system.

A soft link can also be created for directories. For example, the `/bin` directory in Solaris is only a symbolic or soft link to the `/usr/bin` directory. The `/bin` directory itself contains nothing! It only points the user to the `/usr/bin` directory.

Lesson 8.3 Creating Soft Links to a Directory

In this lesson the reader will create a simple text file. Later, a symbolic or soft link will be created to that file.

1. Log in as the root user.
2. Open a Terminal window
To open a Terminal window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Tools menu item, then left click on the Terminal window icon.
3. Type the command `mkdir /linktest`
An ordinary directory named /linktest is created.
4. Type the command `ln -s /linktest /linktest2`
A new directory called /linktest2 is created. This is a symbolic link to the original directory /linktest. The ln command creates this symbolic link.
5. Type the command `ls -l | grep linktest`

The command typed is: **ls -l | grep linktest**. The output of the command should look something like this:

```
drwxr-xr-x 2 root  other   512 Sep  9 17:08 linktest
lrwxrwxrwx 1 root  other    9 Sep  9 17:08 linktest2 -> /linktest
```

This shows the long listing of both the **/linktest** and **/linktest2** directories.

Notice two things about the **/linktest2** listing.

1. The listing starts with a small **L**, which indicates a symbolic link.
2. The last part of the listing shows where the link is pointing (**linktest2 -> /linktest**) Here, **linktest2** is the symbolic link to the **/linktest** directory.

6. Type the command **cd /linktest**

7. Type the command **pwd**

The **pwd** command shows the current working directory. In this example, **/linktest** is an ordinary directory that acts like any other directory on the server.

8. Type the command **cd /linktest2**

9. Type the command **pwd**

As can be seen, the current working directory is **/linktest**. That is because **/linktest2** is a symbolic link to the parent directory **/linktest**. If a user types **cd /linktest2** the user is moved to the **/linktest** directory.

10. Type the command **cd /bin**

11. Type the command **pwd**

Notice that the present working directory is **/usr/bin**. Understand that **/bin** is a symbolic link to **/usr/bin**.

12. Type the command **ls -l | grep bin**

Notice the symbolic link to the **/usr/bin** directory?

```
lrwxrwxrwx 1 root  root    9 Aug 19 14:26 bin -> ./usr/bin
```

Lesson 8.4 Soft and Hard Links to a File

In this lesson the reader will create a simple text file. A hard link will be created to the file and a soft link will be created to the file.

1. Log in as the root user.

2. Open a Terminal window

To open a Terminal window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Tools menu item, then left click on the Terminal icon.

3. Type the command **cd /linktest**

The **cd** command changes the current working directory to the **/linktest** directory.

4. Type the command **echo "link test file"> OriginalFile**

The **echo** command created a simple text file with the name **OriginalFile**.

5. Type the command **ln OriginalFile HardLinkFile**

This command creates a hard link between the original file and the new file **HardLinkFile**.

6. Type the command **ls -l**

```
-rw-r--r-- 2 root  other   15 Sep  9 17:40 HardLinkFile
-rw-r--r-- 2 root  other   15 Sep  9 17:40 OriginalFile
```

Notice that both files are ordinary text files. Neither file is a target or reference to the other file. Both these files **HardLinkFile** and **OriginalFile** are the same. They both point to the same information.

The next five steps show what happens before and after adding text to one of the hard link to the original file **OriginalFile**. Notice that both file's content changes the same.

7. Type the command **cat HardLinkFile**

8. Type the command **cat OriginalFile**

9. Type the command **echo "add another line">> HardLinkFile**

10. Type the command `cat HardLinkFile`
11. Type the command `cat OriginalFile`
The output from the last three lines should look like this:

```
echo "add another line">> HardLinkFile
# cat HardLinkFile
link test file
add another line
# cat OriginalFile
link test file
add another line
```

*Both the **HardLinkFile** and **OriginalFile** hard links point to the same file.*
12. Type the command `ln -s OriginalFile SoftLinkFile`
*The **ln -s** command created a soft link file called **SoftLinkFile** that points to **OriginalFile**.*
13. Type the command `ls -l`
*The **ls -l** command should display the following output:*

```
-rw-r--r--  2 root  other   32 Sep  9 17:45 HardLinkFile
-rw-r--r--  2 root  other   32 Sep  9 17:45 OriginalFile
lrwxrwxrwx  1 root  other   12 Sep  9 17:47 Softlink -> OriginalFile
```
14. Type the command `rm OriginalFile`
15. Type the command `cat HardLinkFile`
*Even though **OriginalFile** was removed, the file's data still exists, because there is still one hard to that data..*
16. Type the command `cat SoftLinkFile`
*Unlike the hard link, the **SoftLinkFile** file is now basically an empty shell, because the parent file that the soft link points to does not exist. So, the soft link is no good.*

What Is a Directory?

A directory is a simple concept—it is just a file that holds file names. The root file (`/`) holds the file names of other files and directories (`/etc` is a subdirectory of the root file system). Each subdirectory under the root file is nothing more than a file that holds files and subdirectories of its own.

What Are Regular Files?

Regular files are files that contain data. The data can be of any type (text, binary, Oracle data, GIF image, and so forth). A file is referenced by a filename (what humans like to work with) and an inode number (what Solaris likes to work with). The file's name is nothing more than a word that points to an inode.

To the operating system, the filename `mydata.dat` is only a word “mydata.dat“ that is referencing an inode, such as 302572. An inode (in this case 302572) contains all the information needed about the file (Created - 2/13/2004, Owner - John Smith, Data on hard drive - block locations 823-825,817, Permissions - salesgroup READ/WRITE, and so forth.) This inode is stored on what is known as the inode table. If a user copies a data file to his or her home directory, that user has in reality copied all the data under inode 302572 to his or her directory. Data is stored in what is known as data blocks. The data blocks are small sections of the hard drive surface. The data IS NOT STORED IN THE INODE ! The inode number only references the data blocks and does not hold the actual data itself.

Using the `ls` Command

The `ls` command is used to tell an administrator what type of file he or she is working with. Each file has a code letter associated with the file. The code letters are:

- c** - Character special file
- b** - Block special file
- d** - Directory
- D** - Door
- l** - Symbolic link
- p** - Fifo or named pipe
- s** - Family socket
- - Ordinary file

Lesson 8.5 Using the **ls** Command

The **ls** command can be used to determine what type of file the administrator is examining. For more information on the **ls** command, type the command **man ls** in a Terminal window.

1. Log in as the root user.
2. Open a Terminal window
To open a Terminal window, right click anywhere in unoccupied desktop space, left click on the Tools menu choice, then left click on the Terminal icon.

3. Type the command **ls -l /etc/passwd**

The output should be something like

```
-r--r--r-- 1 root sys 60 May 6 2002 /etc/passwd
```

This is a regular file, as indicated by the (-) character on the beginning of the file's description.

4. Type the command **ls -l | grep etc**

The output should be something like

```
drwxr-xr-x 41 root sys 3584 May 6 2002 etc
```

This is a directory as indicated by the (d) character on the beginning of the file's description.

5. Type the command **ls -l /dev/dsk/c0t0d0s0**

The output of this command should look something like

```
lrwxrwxrwx 1 root root 38 Sep 30 2001 /dev/dsk/c0t0d0s0 ->
../devices/pci@1f,0/ide@d/dad@0,0:a
```

This is a symbolic link (a soft link name to another file), as indicated by the small L character at the beginning of the file's description.

6. Type the command **ls -l /devices/pci@1f,0**

*Try using the **ls -l** command on different files in the **/devices** file until a character device file is found. A character device file uses the code letter "c" at the beginning of the long listing.*

On the SunBlade 100 owned by the author, the file

***/devices/pci@1f,0/ide@d:devctl** is a character file, as referenced by its long file listing of :*

```
crw----- root sys 135,0 Jul 25 2002 ide@d:devctl
```

This is a character special file as indicated by the small c character at the beginning of the file's description.

Key Points to Remember

A system administrator needs to know the different types of files in Solaris. Use the **ls -l** command and try to find all the different types of files on the server. The most critical files on the Sun server are the files in the **/dev** and **/devices** directory.

Chapter 9 Working with Software Packages

Lessons in This Chapter

Lesson 9.1 Using Command Line Software Package Utilities	9-5
Lesson 9.2 Using the Admintool	9-7
Lesson 9.3 Using the Solaris Product Registry	9-10
Lesson 9.4 Using the Web Start Installer	9-11

Introduction

Most companies purchase and install software on Solaris servers that did not originally come with the server. The software is bundled in what is known as a software package. The software package is a collection of files and directories that make up the software program. There is usually an install script and a post-install script within the software package.

Sun provides several command line utilities to add, check and remove software packages from the system. Sun also provides a GUI (Graphical User Interface) utility to add, check, or remove software packages. Software packages can be installed and maintained by the following software package management command line utilities and GUI tools:

Command line tools	Used to add, monitor and check on software packages when a GUI is not available.
Admintool	An older GUI that was created with Solaris 2.6. It is being phased out of existence with Solaris 9.
Web Start Installer	Also known as “the installer.” A step by step wizard similar to the installation wizards used with Microsoft Windows software packages.
Solaris Product Registry 3.0	A tool that is primarily used to keep registration information. It can also be used to uninstall software packages.

Quick Tip

- Admintool was created with the Solaris 2.6 operating system. When it starts it gives a warning message that essentially says that **Admintool** has been declared obsolete. This means that Sun Microsystems will not officially support **Admintool**.
- Even though Sun does not officially support **Admintool**, it can still be used to install software packages.
- Some later versions of Solaris 9 might not have the **Admintool** included in their CDRoms.

Software Packages

Software applications for SUN Solaris come in what is known as software packages. These are a bundling of all the files and scripts that are needed to install an application on to the Solaris 9 operating system. Sun produces these software packages, as well as third-party software developers. The software packages conform to the ABI (Application Binary Interface) System V specifications. Sun packages use the naming convention **SUNWname**. The first four letters **SUNW** indicate that the package was produced by Sun Microsystems. The last part of the name describes the software package. For example, **SUNWauddx** is a software package produced by Sun. The last part of the name, **auddx**, indicates that this is an audio package that is used with the system's audio devices.

There are two main categories of software packages. The term *Bundled Software Package* refers to software packages that are included with the purchase of the operating system. The term *Unbundled Software Package* refers to a software package that was purchased or acquired outside of the original operating system media.

A Solaris software package consists of different files and directories. The only two important files are:

pkginfo A text file that contains descriptions of the software package. Figure 9.1 is the pkginfo file used with the **SUNW1251f** software package.

```
ARCH=sparc
PKG=SUNW1251f
NAME=Russian 1251 fonts
VERSION=1.0,REV=2001.03.21.17.00
SUNW_PRODNAME=SunOS
SUNW_PRODVERS=5.8
SUNW_PKGTYPE=usr
MAXINST=1000
CATEGORY=system
DESC=Russian additional locale fonts (1251)
VENDOR=Sun Microsystems, Inc.
HOTLINE=Please contact your local service provider
EMAIL=
CLASSES=none Xupdate fontenc
BASEDIR=/
PSTAMP=
SUNW_PKGVERS=1.0
PKG_SRC_NOVERIFY= none
```

```

PKG_DST_QKVERIFY= none
PKG_CAS_PASSRELATIVE= none
#FASPACD= none

```

Figure 9.1 SUNW1251f pkginfo File

pkgmap

A text file that gives control information to the Solaris 9 software management tool. This gives information such as what files to copy, what directories to create, and what permissions to set. This is a great file to troll for information on what the software package contains. Figure 9.2 is a screen shot of the pkgmap file associated with the **SUNW1251f** software package.

```

1 d none usr 0755 root sys
1 d none usr/bin 0755 root bin
1 f none usr/bin/lp_1251 0755 root bin 172 14147 950806664
1 d none usr/openwin 0755 root bin
1 d none usr/openwin/bin 0755 root bin
1 f none usr/openwin/bin/mp_1251 0755 root bin 84 6918 950806614
1 d none usr/openwin/lib 0755 root bin
1 d none usr/openwin/lib/X11 0755 root bin
1 d none usr/openwin/lib/X11/fonts 0755 root bin
1 d none usr/openwin/lib/X11/fonts/TrueType 0755 root bin
1 d none usr/openwin/lib/X11/fonts/TrueType/ttmap 0755 root bin
1 f none usr/openwin/lib/X11/fonts/TrueType/ttmap/ansi-1251.ttmap 0644 root bin

```

Figure 9.2 SUNW1251f pkgmap File

Software Management Command Line Utilities

Table 9.1 shows the command line utilities to add, check, and remove software packages:

Command Line Utility	Description
Pkgadd	Adds a software package
Pkgrm	Removes a software package
Pkgchk	Checks if a software package is installed correctly
Pkginfo	Displays information on a software package
Pkgask	Saves answers to an install in a response file, which can be used to automate batch installs
Pkgparam	Displays parameters associated with a software package

Table 9.1 Command Line Package Management Tools

These command line utilities can be used in scripts to automate software package installation. They are also necessary for software management on a server when the GUI is not available. Remember, if there is a severe problem booting Solaris 9, it will halt at the Single User Run level. There is no GUI available at this level.

The pkgadd Command

The **pkgadd** command is used to add software packages to the system. It reads the software package, uncompresses files, and copies the files to the appropriate location. It also saves package installation information in the **/var/sadm/pkg** directory. The **/var/sadm/pkg** directory holds directories that look like Figure 9.3 (depending on what packages have been installed).

```
ls /var/sadm/pkg
```

```
SUNW1251f  
SUNW1394h  
SUNW1934x  
SUNWjdr
```

Figure 9.3 Contents of the /var/sadm/pkg directory

To add a software package to the system, first change to the directory that holds the software package. For example:

```
cd /cdrom/cdrom0/<directory_holding_software>
```

Then type the command

```
pkgadd -d <software_package_name>
```

To view information about a software package without installing the package use the **-l** (little L) option. For example:

```
pkgadd -d /absolute/path/of/software/directory/ -l SoftwarePackage
```

The pkgrm command

The **pkgrm** command is used to properly remove software packages from a Solaris 9 system. It is a very bad idea to use the **rm** command to remove directories associated with a software package. If a software package is improperly removed, a senior level system administrator would then have to look through the original software package contents and remove each file by hand. Other system modifications would also have to be made. For example, processes could still be running from the original software package. Another potential problem is that run control scripts that have not been removed could hang when trying to start software programs that no longer exist. This could hang the boot process and crash the server. The key point is to always use the proper software package management tools.

To remove a software package, simply type the command:

```
pkgrm <software_package_name>
```

For example:

```
pkgrm SUNWauddx
```

```
pkgrm -s /path/directory/<software_package_name>
```

The **-s** option tells the **pkgrm** command to remove the software package from a specific directory.

The pkgchk command

The **pkgchk** command is used to check if a software package has been installed correctly on a server. This command compares the installed files with the software package. For the most accurate check, run this command immediately after installing the software. If this command is run days or weeks after the original installation, the results can be somewhat misleading, because the files could have changed due to the ordinary use of the program. In that case, take its analysis with a grain of salt. If the software program is running correctly, leave it alone.

To run **pkgchk** type the command:

```
pkgchk <software_package_name>
```

for example:

```
pkgchk SUNaudio
```

The **pkgparam** command

The **pkgparam** program shows detailed information on a software package. This program supports the following options:

```
pkgparam <options> <software_package_name>
```

<options>

-a Compare only the file's attributes. The attributes refer to the read/write/execute permissions on a file. Every file and directory can have three types of permissions set: read, write and execute. These permissions can be given to the user that owns the file, other people in a specified group, and to other users. The **-a** argument tells **pkgparam** to check the file's permissions against the original file permissions set up during the installation.

-c Check the file's contents only.

-d <directory>

Specifies the path to the spool directory. If a software package is spooled to the hard drive, the **-d** option can check the spooled version of the software with the original version. This is a very good check to perform. The file's contents and permissions should be the same between the installation source and the spool directory.

-v Verbose mode, display information for every file that is being checked.

An example of checking a software package:

```
pkgparam -v SUNWauddx
```

Lesson 9.1 Using Command Line Software Package Utilities

The software package **SUNW1251f** (Russian 1251 fonts) will be used extensively during this lesson. It is a very useful software package because if it is incorrectly added or removed it will not harm the system. This lesson will cover several command line software package utilities. Make sure the workstations involved are capable of displaying the CDE (Common Desktop Environment).

1. Log in as the root user.
2. Open a Terminal window
To open a Terminal window, right click anywhere in unoccupied desktop space, left click on the Tools menu item and then left click on the Terminal icon.
3. Type the command **cd /cdrom/cdrom0/s0/Solaris_9/Product**
The /cdrom/cdrom0/s0/Solaris_9/Product directory contains information on all the software packages installed on the system.

4. Type the command **pkginfo -l SUNW1251f**
*This should give information about the **SUNW1251f** software package.*
5. Type the command **pkginfo**
This shows all the packages installed on the system.
6. Type the command **pkginfo | grep SUNW1251f**
*This command uses **grep** to show only the information about the Russian fonts that are installed on the server.*
7. Type the command **pkgrm SUNW1251f**
*Before removing a program, **pkgrm** asks twice **Are you sure you want to continue?** Answer **Y** for both of these questions.*
8. Type the command **pkgadd -d /cdrom/cdrom0/s0/Solaris_8/Product**
*This command scans the specified directory and then asks what packages should be installed. Press **CTL + C** to not select any software packages. The next step adds only the Russian fonts.*
9. Type the command **pkgadd -d /cdrom/cdrom0/s0/Solaris_9/Product SUNW1251f**
This command adds the Russian fonts back on to the server.
10. Type the command **pkginfo | grep SUNW1251f**
This command verifies that the Russian fonts were re-installed on the server.
11. Type the command **pkgchk SUNW1251f**
*This command verifies the installation of the Russian fonts. The **pkgchk** only performs a crude byte count, so it may report errors even after a fresh install—it's far from perfect!*
12. Type the command **pkgadd -d /cdrom/cdrom0/s0/Solaris_8/Product -s /tmp SUNW1251f**
*This command spools (copies) the source code of the Russian fonts to the **/tmp** directory. It does not install the software package; it only copies the source files onto the server.*

Spooling a Software Package

Sometimes, it may be necessary to install the same software package onto multiple servers. In this case, it might be convenient to spool the package onto a hard drive, rather than using a CDROM each time. By spooling packages onto a fast hard drive, you can install more quickly than from a slow CDROM. You can then use the **pkgadd** command to copy files from the central spool location onto other servers.

The default spool location is **/var/spool/pkg**. When a software package is spooled to this location, the system administrator only needs to type the command **pkgadd <package_name>**. This is convenient because the administrator does not need to specify the location of the package. The **pkgadd** command checks the **/var/spool/pkg** directory first.

For example, type:

```
pkgadd -d /cdrom/cdrom0/path_to_software_package -s /spool/mydirectory  
<package_name>
```

This command will spool a software package to the destination directory **/spool/mydirectory**

If a spool directory is not specified after the **-s**, option, the **/var/spool/pkg** directory will be used by default, as in

```
pkgadd -d /cdrom/cdrom0/packages -s SUNWauddx
```

This command would save the software package **SUNWauddx** to the **/var/spool/pkg** directory.

Now, if you type the command

`pkgadd SUNWaudx`

the audio package would be installed.

Using GUI Software Package Management Tools

Sun provides two GUI tools to use with software package management. The first tool is the Admintool. It is a GUI based tool that is going to be phased out in the near future. The other tool is the Solaris Product Registry that supports the GUI installers that are found on modern CDRoms.

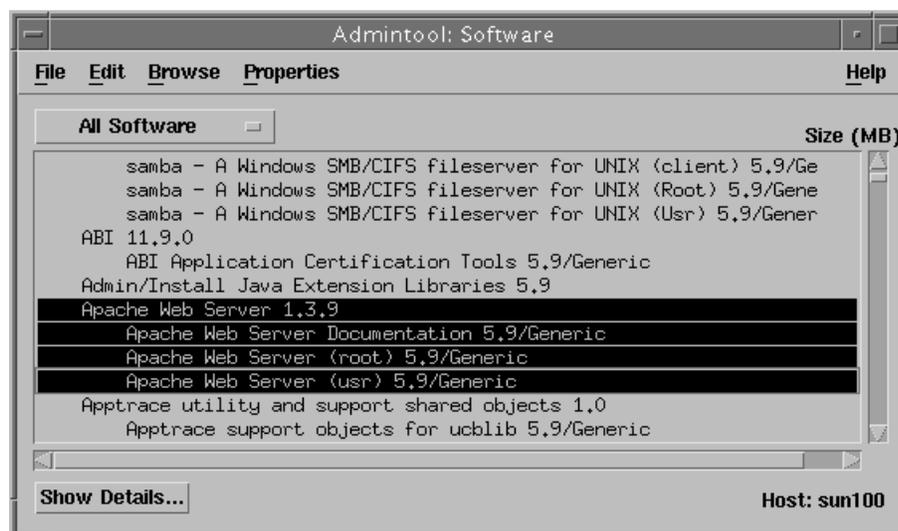
Admintool

This application shows software packages, checks on software packages, removes software package. To use admintool to add, check or remove software packages you must be the root user or a user in group 14, the sysadmin group. Type the command `admintool &` to start the Admintool. The ampersand (&) symbol on the back of the command tells Solaris to run the program in the background. This gives you control over your terminal window again. Do not close the Terminal window or Console window that the command was type in or the application will die when the window closes.

Lesson 9.2 Using the Admintool

This lesson covers the use of the Admintool for software package management. This exercise requires the use of the Software 1 of 2 CDRom.

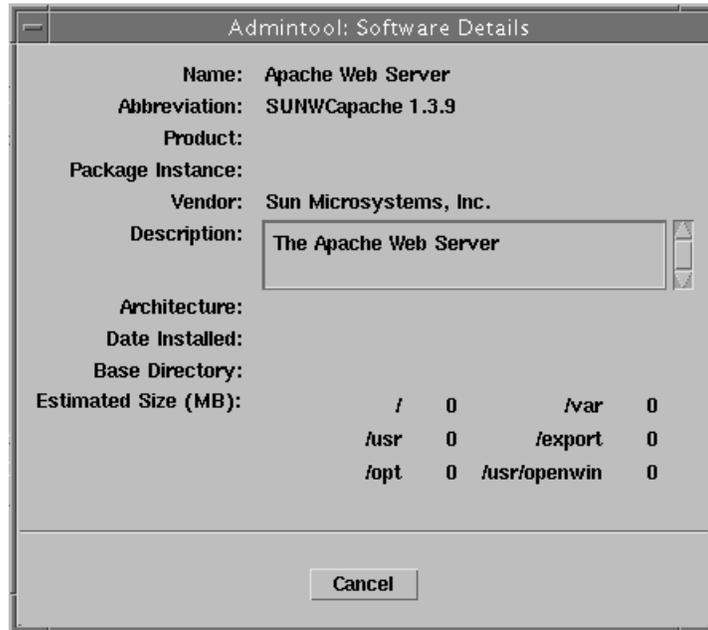
1. Log in as the root user.
2. Open a Terminal window
To open a Terminal window, right click anywhere in unoccupied desktop space. Left click on Tools, then left click on the Terminal icon.
3. Type the command `admintool &`
4. Left click on Browse on the menu bar.
5. Left click on Software from the drop down menu.



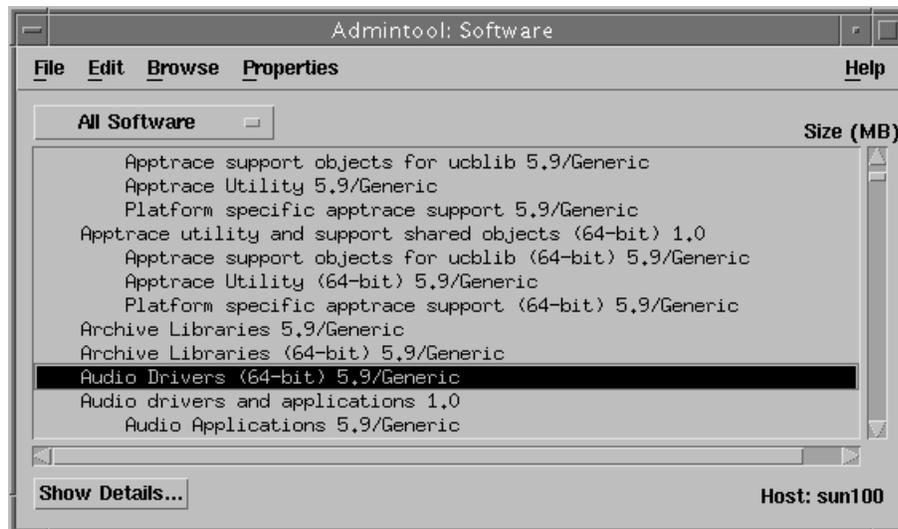
If the system software is not displayed, select Software from the Browse menu item at the top of the window.

6. Select the **Apache Web Server 1.3.9**

- Click on the **Show Details...** button (located on the lower left corner)

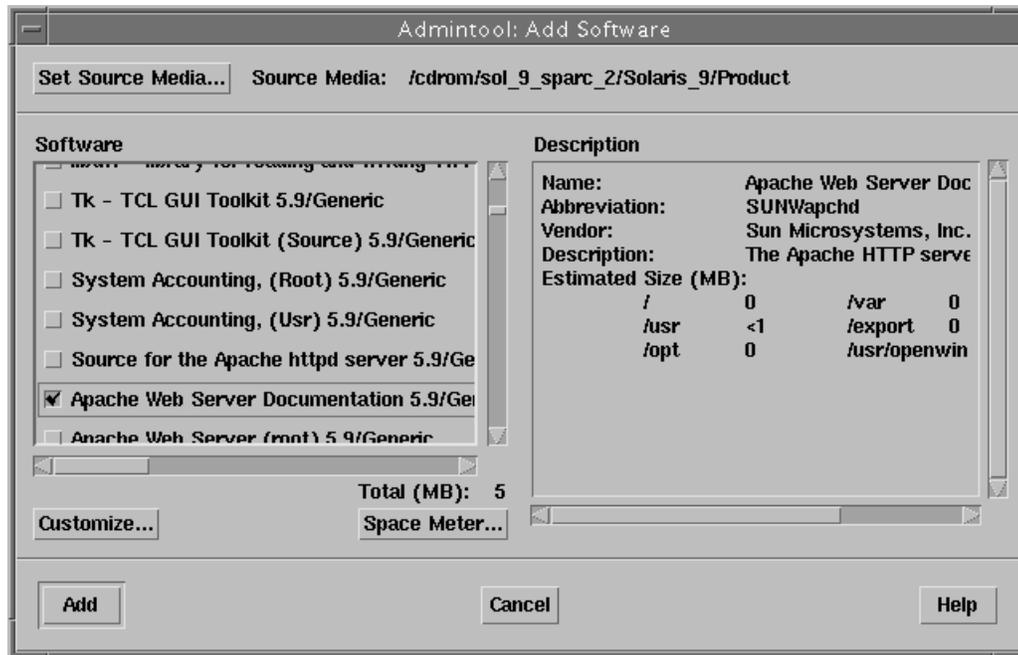


- Click on the **Cancel** button.
- Scroll down and highlight the **Audio Drivers (64-bit) 5.9/Generic** package



- Click on the **Edit** menu bar option.
- Click on the **Delete** pull down menu option.
- Do not actually delete the software package! Instead, click on the **Cancel** button
- Click on the **Edit** menu bar option.
- Click on the **Add...** menu item
A small popup window may appear. This popup window will ask for the location of the software package source. Type in the path `/cdrom/cdrom0/Solaris_9/Product`. Make sure that the Solaris Software 2 of 2 CDROM is in the CDROM drive).

15. Click on the **OK** button.
This could take some time, so be patient.
16. A new window (as shown below) will appear. Scroll down the right view pane and select **Apache Web Server Documentation 5.9/Generic**.



*When the **Apache Web Server Documentation 5.9/Generic** is selected, a description of the software package will be shown in the right view pane.*

17. (This is an optional step.) Left click on the **Add**, **Customize...** and **Space Meter...** buttons in this window.
Play around with the windows that are displayed after clicking on these buttons. Just do not add or remove any software from the system without knowing the consequences of these actions.
18. After playing with Admintool, left click on **Cancel** until you have exited from the Admintool.

The Solaris Product Registry

This software management tool is used with some of the latest software packages that have been created for Solaris. These packages come with Microsoft Windows type GUI install wizards. These are known as “Web Start 3.0” wizards. You can view the installed software packages, find and run an installer for a package, and add and remove software packages with the Solaris Product Registry.

Solaris Product Registry is a GUI utility that displays what software packages have been installed on a system. It is 100% compatible with the **pkgadd** command, the Solaris Web Start Program and the Admintool.

To run Solaris Project Registry, give the command

```
/usr/bin/prodreg &
```

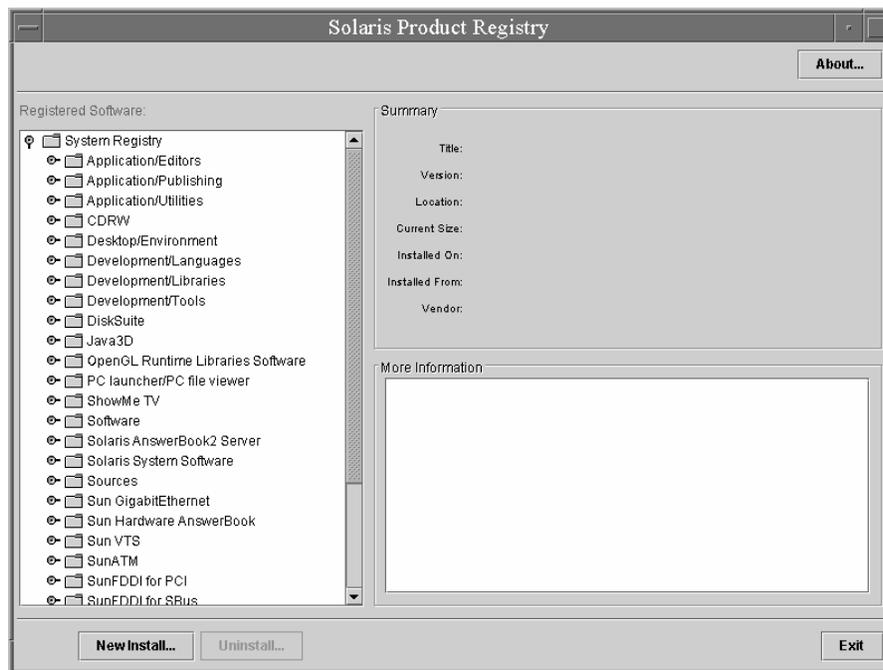
only the root user can work with this utility.

Solaris Product Registry does not support administration files. Administration files are used to automatically answer the questions like “Are you sure?” and “Location to install the software package?” presented during the installation of software.

The Solaris Product Registry 3.0 does not allow the administrator to spool software packages onto the hard drive. However, spooling a software package to a directory can be done by hand or by using the **pkgadd -s** command. Just copy the contents of the software package or CDROM to the hard drive. The software package can then be used as a central repository for network installations.

Lesson 9.3 Using the Solaris Product Registry

1. Log in as the root user
2. Open a Terminal window
To open a Terminal window, right click anywhere in unoccupied desktop space. Left click on Tools then left click on the Terminal icon.
3. Type the command **/usr/bin/prodreg &**



4. In the screen shown above, highlight **ShowMe TV**
The ShowMe TV software package is rather obscure software package that will not damage the system if it is removed, damaged or added to a server.
5. Click on the **Uninstall...** button

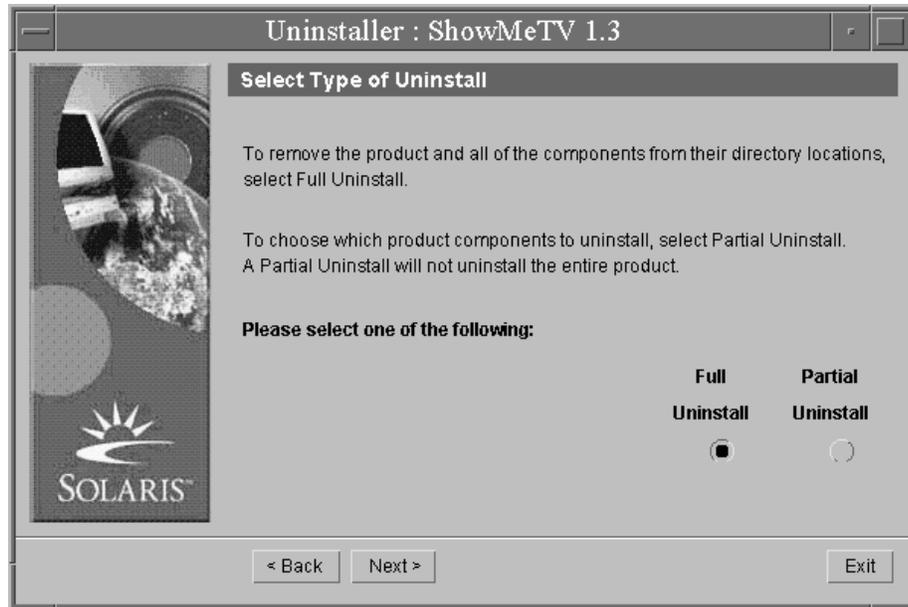


Figure 9. 4 Uninstaller for ShowMe TV

6. Click on the **Exit** button on the bottom to cancel the uninstall.

There is no reason to uninstall ShowMeTV 1.3

Solaris Web Start Program

The Solaris Web Start program can be run after the operating system installation to install additional products from the Solaris 9 Media package. When a Solaris 9 CDROM is inserted into the CDROM drive, the File Manager should automatically start. The File Manager displays the Installer.



To start a Web Start Installation, left double click on the Installer icon.

You can also run Web Start from a Terminal window or Console window. To do so, type the command `/<absolute/path/to/directory>/installer &`.

To run Web Start without a GUI display, type the command `installer -nodisplay`. The installer also supports the `-noconsole` argument that allows the software being installed to automatically install in its default configuration.

Unlike the command line utilities, Solaris Web Start does not allow for the creation of an administration file. An administration file is used to automate the installation of a software package by custom answering yes/no questions and directory location questions. Also, this tool does not allow the administrator to spool software packages to a spool directory. That is not a big issue, because it is just as easy to copy the contents of a CDROM to a directory by hand. The Solaris Web Start Program does not allow for the installation of individual software packages, only software clusters..

Lesson 9.4 Using the Web Start Installer

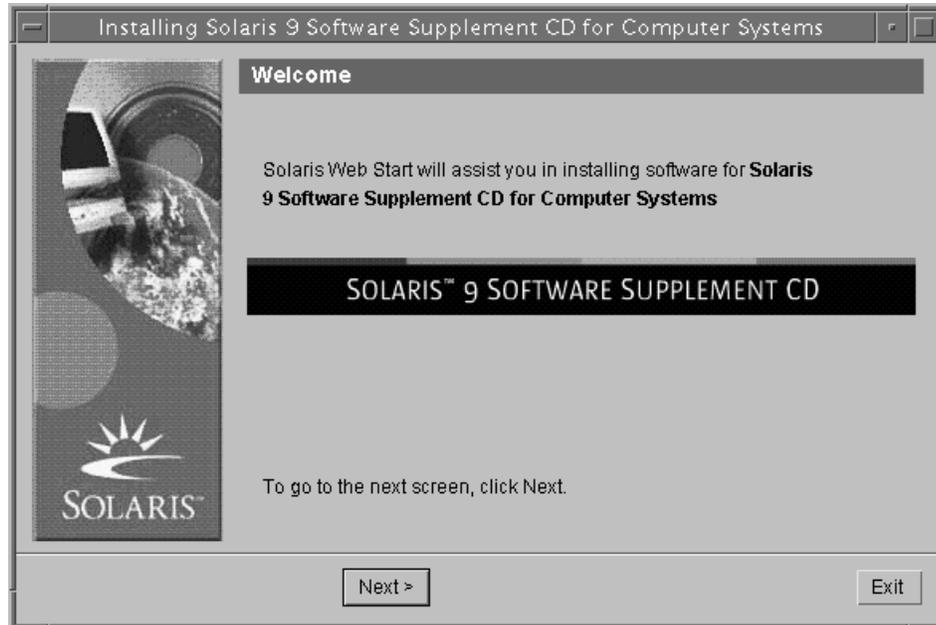
This lesson requires the use of the Solaris 9 Software Supplement CDROM. This CDROM comes with all versions of the Solaris 9 media kits.

1. Log in as the root user.

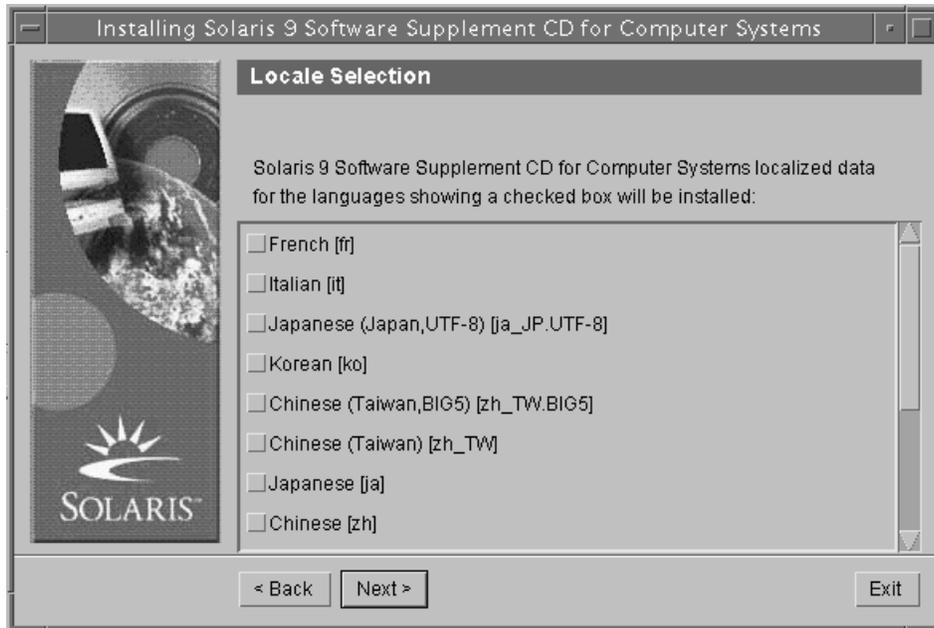
2. Insert the Solaris 9 Software Supplement CDROM into the CDROM drive.
*If the CDROM drive has a CDROM currently inside it, it will not open. Type the command **eject cdrom** in a Terminal window to eject the current CDROM. It is also possible to eject the current CDROM by using the File Manager utility. Select **File**, then select the menu item **eject**.*

When the Solaris 9 Software Supplement CDROM is inserted into the CDROM drive the File Manager should automatically start.

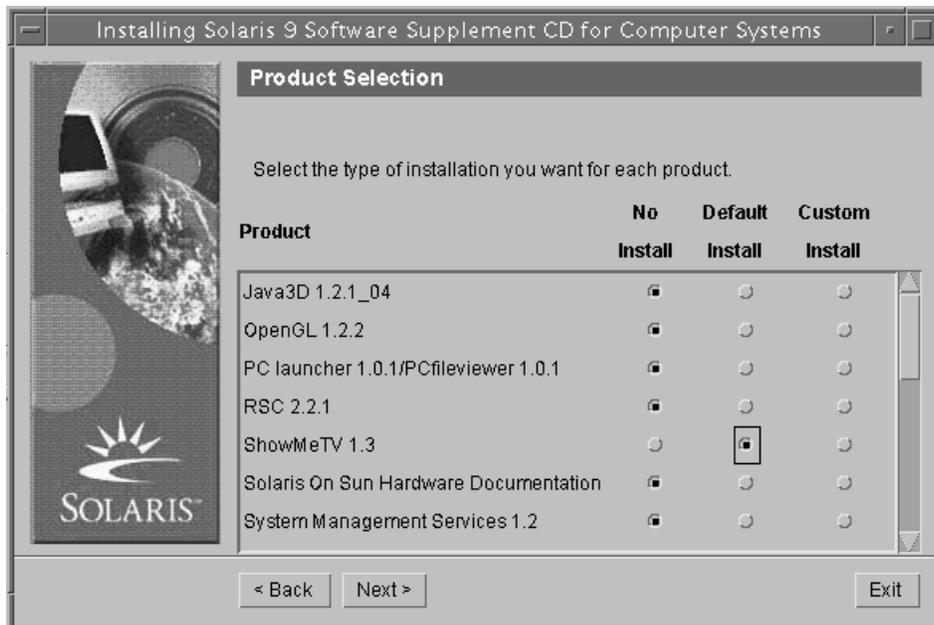
3. Left double click on the Installer icon.
The Solaris 9 Software Supplement CD splash screen should appear next.



4. Left click on the **Next >** button
You will see a screen that shows various optional languages that can be used for the installation.
5. Do not choose any of these Locales—just click on **Next >**



The next screen shows the optional software components that can be installed with Solaris 9.



6. For this lesson, make sure to select **No Install** for all the components except **ShowMeTv.1.3**. For the program **ShowMeTv.1.3** check the **Default Install** checkbox.
7. Left click on the **Next >** button
The next screen is a confirmation screen for the ShowMeTV 1.3 Software package.



8. (Optional): If you want to install ShowMe TV 1.3, left click the **Install Now** button.
9. To exit from Web Start Installer, left click the **Exit** button.

Key Points to Remember

A system administrator needs to know how to add and remove software packages. Understand that most companies do not have a GUI display on a server. The **pkg...** commands have to be mastered by a system administrator. The Solaris 9 software CDROMs have all the software packages that could be added during the installation of Solaris 9.

Chapter 10 Working with Patches

Lessons in This Chapter

Lesson 10.1 Understanding the <code>/var/sadm</code> Directories.....	10-3
Lesson 10.2 Installing a Single Patch.....	10-4
Lesson 10.3 Removing Patches.....	10-6

Introduction

This chapter requires the reader to download the latest copy of the Solaris 9 recommended patch cluster. This can be found at the website <http://sunsolve.sun.com>. For demonstration purposes it's best to install that single patch first, then install the Solaris 9 recommended patches later. The key is to see how Solaris 9 installs a single patch, and then install the patch cluster. This single patch can be any recent patch for Solaris 9 SPARC. The patches can then be burned onto a standard CDROM on a Microsoft Windows system.

If for some reason the filename for a patch becomes abbreviated with Microsoft's old 8.3 naming pattern (example: **My Documents** is renamed **mydocu~1**) just save the patch with an 8.3 file name (like **patch.sep**). After the patch is copied to the SUN workstation, restore the file's name back to its original condition. Microsoft's 8.3 naming convention in the year 2002 is a very annoying throwback to MS-DOS that should have been discarded a long time ago.

Why Have Software Patches?

No large scale software program is written and debugged perfectly the first time. Eventually, end users and testers find problems with a software package after it has been released. The developers of these software packages then create what is known as a patch. A patch is nothing more than a collection of replacement files that fix problems in the original software package. Sometimes a patch is used to add new features to a software package that could not be added in time for the initial software rollout.

Sun software patches are given serial numbers for identification. Sun patches are broken down into two sections, a patch number and a revision number. For example, you might receive patch 103426-03.

103426	The patch number
03	The revision number of the patch.

There are three main sources to get sun patches:

- <http://sunsolve.sun.com> is the webpage
- Anonymous FTP from sunsolve.sun.com (log in as "anonymous" and use your email address as the password). The public patches are in the `/pub/patches` directory
- Patch Update CDROMs that are released every two months

Corporations with SunService contracts can use the private section of sunsolve.sun.com to download patches that are not distributed on the free section of the [sunsolve](http://sunsolve.sun.com) site.

The patches are distributed in a zip file named after the patch number. For example, **103426-03.zip** would be the file name for the 103426-03 patch. The command

```
unzip 103426-03.zip
```

would be used to extract all the files out of the zip file. This will create a directory named 103426-03 on the hard drive.

It is very important to understand that the patches must be uncompressed before these utilities will work. Table 10.1 shows some command line utilities that work with patches.

Patch Command	Description
showrev -p	Shows all the software patches on the system
patchadd -p	Shows all the software patches on the system
patchadd -d <patch_number>	Adds a patch without using the <code>/var/sadm/patch</code> back out directory (doing it this way is not a good idea)
pkgparam <software_package_name>	Shows all the patches applied to a specific software package
PATCHLIST	Shows information on a patch added to a software package
pkgparam <software_package_name>	Shows information on a patch added to a software package
PATCH INFO <patch_number>	Shows information on a patch added to a software package
patchadd -R client	Shows all patches applied to a client
patchadd <patch_number>	Adds a specific patch
patchrm <patch_number>	Removes a patch
patchrm -f <patch_number>	Forcefully removes a patch

Table 10.1 Command Line Patch Tools

The patchadd command

The **patchadd** command is used to apply patches to the operating system or a to net install image. This command must be run by the root user. Logging in as the root user is necessary so that a hacker can not abuse the patch process by replacing valid files with worm files or virus-based files.

The **patchadd** command supports the following options:

```
patchadd <options> patch#
```

-B </backout/directory>

Specifies a specific backout directory instead of the default backout directory.

-C <net install image>

Apply the patch to the net install image. The net install image is an image of the operating system installation CDROMs that is shared on a network. When the **-C** option is used, the patches replace the same files on the install image. When the new servers are built from the net install image, some patches will not need to be installed.

-d The **-d** option does not create a backout directory. This is a very bad idea. Sometimes the patch itself creates a problem and needs to be backed out (uninstall the patch).

- M** `</patch/directory> <patch# or patchfile>`
Specifies the path and patch number to be installed. This requires the root user specify the absolute path of the directory that contains the patch. The patch number(s) must be entered for each patch. A list of patches in the patch directory can be used instead of typing each patch number separately.
- p** Shows the patches that are currently installed on the system.

The `/var/sadm/patch` Directory

The directory `/var/sadm/patch` contains a list of all the patches on the system. This directory also contains the files that were replaced by the software patch. If a patch needs to be “rolled back” or removed, the previous files can be restored from this directory.

The `/var/sadm/patch` directory has two important types of subdirectories:

`/var/sadm/patch/patch#` This subdirectory is named after a patch. For example, the author’s computer has a subdirectory called

`/var/sadm/patch/108528-09`

The directory contains three text files:

- log** A log file of the patch installation (was the patch installed OK, were any files missing....).
- README.108528-09** A file with special warnings and error messages, patch creation date, OS version, and what bugsIDs were fixed with the patch.
- postbackout** A shell script that is used to clean up the system if the patch is removed

The other important sub-directory is :

`/var/sadm/pkg/PackageName/save/patch#`

For example, the author's computer has a subdirectory called

`/var/sadm/pkg/SUNWtnfcx/save/108528-09`

The `108528-09` subdirectory has a file named `undo.Z`. This compressed `.Z` file has all the original files from the original `SUNWtnfcx` software package that were replaced. If the patch was removed, this original files would be restored from the `undo.Z` file.

Lesson 10.1 Understanding the `/var/sadm` Directories

The `/var/sadm` directory contains two types of subdirectories that are very useful for understanding how patches work. This lesson will guide readers through various `ls` and `cd` and `more` commands to explore these subdirectories.

1. Log in as the root user.

2. Open a Terminal window
To open a Terminal window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Tools menu item, then left click on the Terminal icon.
3. Type the command **cd /var/sadm**
Notice the patch and pkg directories.
4. Type the command **cd patch**
This changes the current working directory to the /var/sadm/patch directory.
5. Type the command **ls**
The /var/sadm/patch directory contains a symbolic directory for every patch installed on the system
6. Type the command **cd <patch#>** (for example, **cd 108528-09**)
7. Type the command **ls**
There should be two or three files (log, postbackout and README.108528-09)
8. Type the command **more log**
The command more displays large text files one screen at a time.
9. Type the command **more README.<patch#>**
This command displays the README file associated with a patch.
10. Type the command **more postbackout** (this file might not exist)
11. Type the command **cd /var/sadm**
12. Type the command **cd pkg**
13. Type the command **ls**
This directory contains a subdirectory for every software package installed on the system.
14. Type the command **find /var/sadm -name <patch#>**
This command shows where all the directories are with the patch# in the /var/sadm directory structure. Write down the locations of the patch# directories.
15. Type the command **cd /var/sadm/pkg/<pkg name>/save/<patch#>**
(for example, cd /var/sadm/pkg/SUNWcpc/save/108528-09)
16. Type the command **ls**
This should show just one file: undo.z. This file contains the files and directories that were replaced by the patch.

Lesson 10.2 Installing a Single Patch

This lesson covers the installation of a single patch and the “9_Recommended” patches. Patches can be installed one by one, or they can be installed in what is known as a “patch cluster” or “recommended patches.” This lesson will install a single patch then a patch cluster to the system.

1. Log in as the root user.
2. Open a Terminal window
To open a Terminal window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Tools menu item, then left click on the Terminal icon.
3. Type the command **mkdir /tempdir**
This will be a temporary directory for downloading. Download the latest recommended patch cluster to this directory.
 - The download page is <http://www.sun.com/software/download>.
 - The patch name should be **9_Recommended.zip***Download this file to the /tempdir directory.*
4. Unzip the patch cluster with the command **unzip 9_Recommended.zip**
*Use the command **pkunzip 9_Recommended.zip**. (If for some reason Sun changes the filename, use the command **pkunzip <cluster-name>.zip** to unzip the file.) A subdirectory should now exist with the name **9_Recommended** (or whatever cluster name was used).*
5. Type the command **cd 9_Recommended**
6. Type the command **ls**

There should be several patches listed, and the following text files.

CLUSTER_README	Information about the patch
copyright	Copyright legal information
install_cluster	Script to automate the installation of the patch
patch_order	Text file used to control what patches are installed first

7. Type the command **cat patch_order**
Take special note of which patch is listed first. The **patch_order** file is a configuration file used by the software patch installation program. Write down the first patch that is listed in this file.
8. Type the command **patchadd <first_patch_listed_in_the_patch_order_file>**
(for example, **patchadd 112875-01**).
This command installs the patch. The process can be rather slow, so be patient.

The eventual output should be:

Checking for installed patches...

Verifying sufficient file system capacity (dry run method)...

Installing patch packages...

Patch number <patch#> has been successfully installed.

see **/var/sadm/patch/<patch#>/log** for details

Patch packages installed:

SUNWaccu

9. Type the command **/tempdir/9_Recommended/install_cluster**
The **install_cluster** utility is used to install all the patches in the software cluster.
10. Answer the question **are you ready to continue with install? [y/n] y**
The message **Installation of <patch#> failed. Return code 2** is not serious. This only indicates the current patch can not be installed due to a previous patch. That is common with a patch cluster.
Go take lunch or a 1 hour break! Seriously!

The patchrm Command

The **patchrm** command is used to remove a software patch from the system. The command **patchrm -f <patch_number>** removes a patch forcefully if there is a major problem that will not allow the software patch to be removed by ordinary methods.

patchrm can not be used if one of the following conditions exists:

- The patch was made obsolete by a newer version of a patch.
- The patch is required by another patch on the system.
- The original software package has been removed.
- A "patch back-out directory" was not created when the original patch was installed (if the software package was originally installed with the command **patchadd -d <patch_number>** to install the patch. The patch can not be removed because the original files were not saved in a back-out directory.

Patch Reports

Sun Microsystems distributes a text file named **Solaris 9 Patch Report**. Most patches also come with a patch report specific to that patch. The Patch Report is a summary report that keeps system administrators updated on the latest patches produced by Sun.

The Patch Report can be downloaded from several locations on the <http://www.sun.com> webpage. The easiest place to find the report is on the webpage <http://sunsolve.sun.com>. This website is the main patch/troubleshooting website for Sun Microsystems. There is also a summary report, **Solaris 9 Patch Report**, that

summarizes all the current patches available for the Solaris 9 operating system. The Patch Report has the following sections:

- New Patches Released Since Last Report
- Update Revs Released Since Last Report
- Solaris 9 Recommended Patches
- Solaris 9 Patches Containing Security Fixes
- Solaris 9 Patches Containing Y2000
- Solaris 9 Obsolete Patches
- Solaris 9 Complete Listing of Released Patches

Patch Structure

A patch has a structure that follows this basic design:

Patch Directory	110342-03	/ install.info	text file
		/ README.110342-03	text file
		/ SUNWsdfr	software package
		/install /copyright	text file
		/scripts	text file
		/pkginfo	text file
		/pkgmap	text file
		/reloc	files and directories used to replace bad files

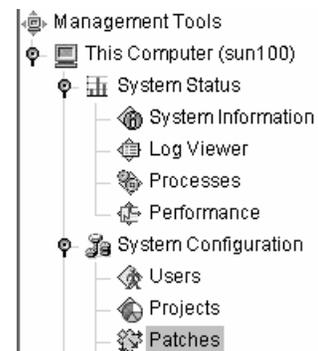
Using Patch Manager

Patch Manager is a GUI utility used to manage patches on the host system or on foreign systems. It runs under the Solaris Management Console 2.1. This utility lets the system administrator add and remove patches from the host system or foreign systems. The patch properties can also be viewed with this tool.

Lesson 10.3 Removing Patches

In this lesson a patch will be removed and then added back again with the Patch Manager tool. Details about the patch will also be explored with the Patch Manager tool.

1. Log in as the root user.
2. Open a Terminal Window
To open a Terminal window, right click anywhere in unoccupied desktop space. The Workspace menu should appear. Left click on the Tools menu item, then left click on the Terminal icon.
3. Type the command **patchrm 112785-05**
*If the patch 112785-05 is not on the workstation, choose a different patch to remove. **BE SURE TO WRITE DOWN THE NAME OF THE PATCH YOU ARE REMOVING!***
4. Start the Solaris Management Console
*To start the Solaris Management Console, right click anywhere in unoccupied desktop space. Left click on Tools, then left click on the **Solaris Management Console** icon.*
5. Left click on the **This Computer (sun100)** icon
6. Left click on **System Configuration**
7. Left click on **Patches**

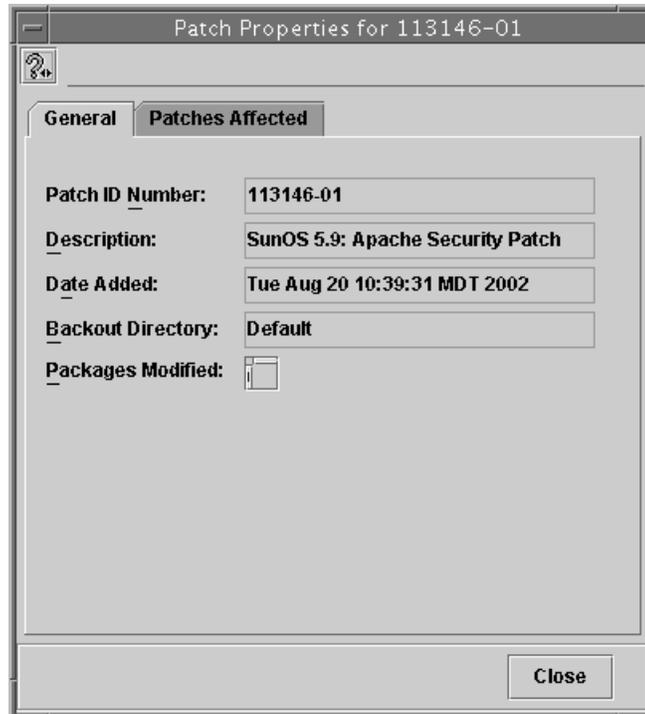


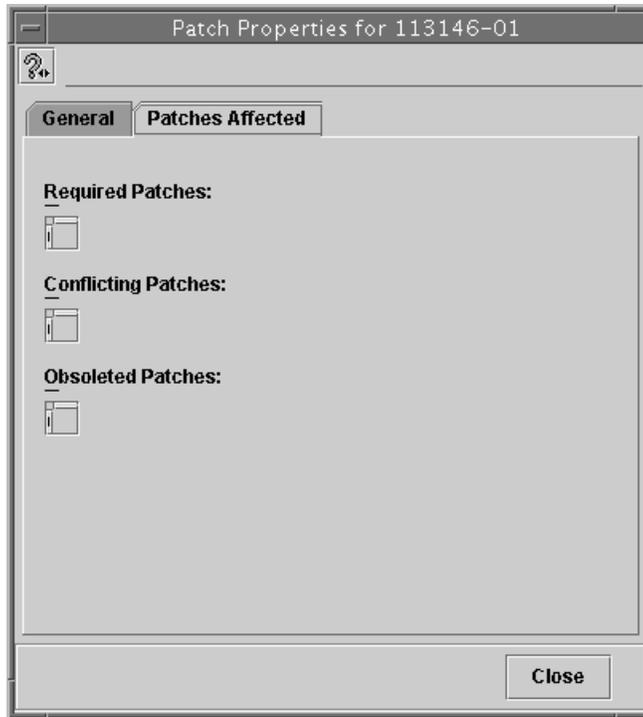
The current system patches will appear in the right view pane.

112785-05	X11 6.6.1: Xsun patch
112875-01	SunOS 5.9: patch /usr/lib/netsvc/rwall/rpc.rwall
113068-01	SunOS 5.9: hpc3130 patch
113146-01	SunOS 5.9: Apache Security Patch

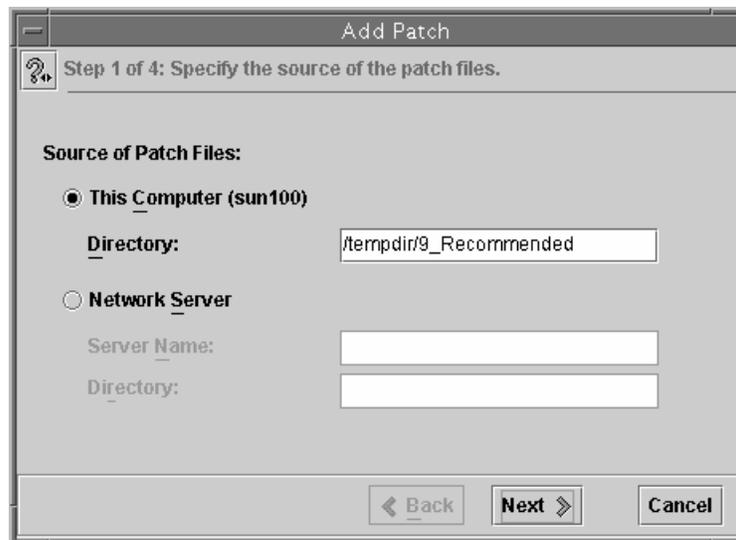
8. Left double click on a patch.

The properties of the patch and the patches that this patch affects are shown in the following two snapshots. Unfortunately Sun only had four patches for Solaris 9 at the time this book has been written, so the results are less than dramatic.

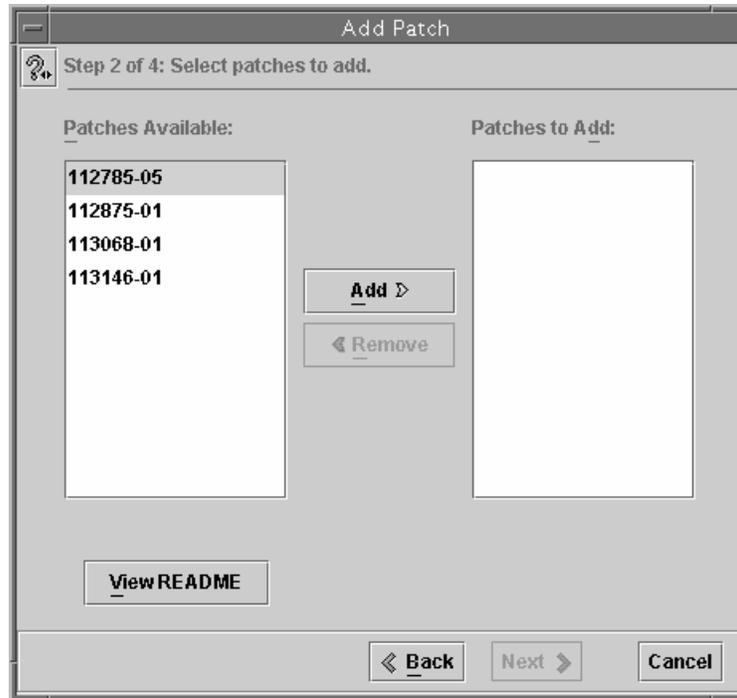




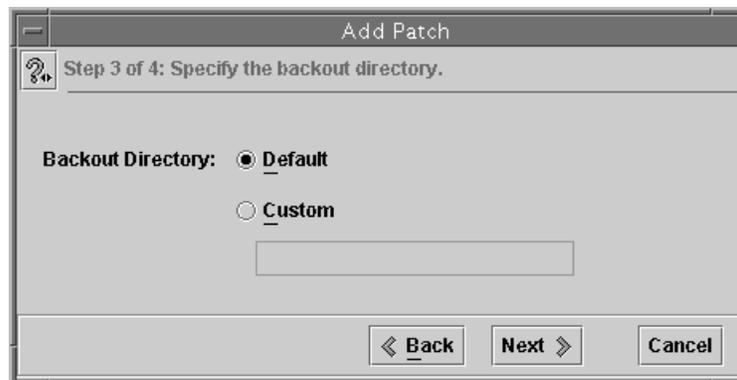
9. left click on the **C**lose button
10. Left click on the **A**ction Menu bar item.
11. Left click on **A**dd Patch...
12. Select **T**his **C**omputer (**s**un100) and type in the path `/tempdir/9_Recommended`



13. Left click on the **N**ext > button

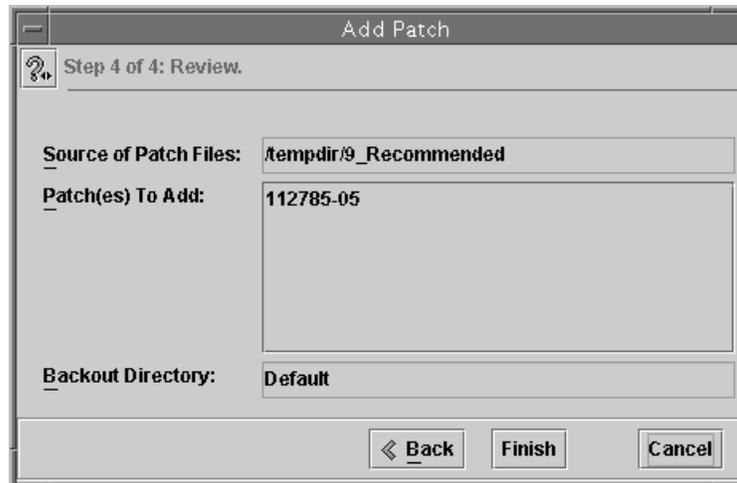


14. Highlight a patch and then left click on the **View README** button to read information about a patch.
15. Left click on the Close button when finished.
16. Highlight a patch in the left view pane then left click on the Add > button
17. Left click on the Next > button when it becomes available.



Make sure the Backout Directory is the Default directory.

18. Left click on the **Finish** button.



Key Points to Remember

This chapter described how to install patches on the system. Security patches will come in from time to time. Make sure to check the Sun websites about once a week. Hackers will watch the Sun websites on a daily basis to learn what security holes have just been discovered. If a company's production servers are not properly patched, a hacker can easily break into the servers. On the other hand, software developers write their code on a set patch level (patches from a given date). There are times when a patch is created that solves one problem but creates bigger problems with other parts of the server. Welcome to the wonderful world of Solaris patches!

Chapter 11 Job Control

Lessons in This Chapter

Lesson 11.1 Create a Simple Cron Job	11-3
Lesson 11.2 Using the <code>at</code> Command	11-5
Lesson 11.3 Using the Process Manager	11-6
Lesson 11.4 Using the <code>prstat</code> Command.....	11-7

Introduction

There are times when a system administrator would like to run a command late in the evening or during other off-hour times without being present. Solaris 9 provides two different utilities to run commands at pre-set times.

The `crontab` command and the `at` command are the primary tools used to perform this task. The `crontab` utility is used to run a command on a regularly scheduled basis. If a system administrator wanted to run the backup command every Sunday night at 1:00 a.m. he or she would most likely use the `crontab` command.

The `at` command is used to run a command only once. Without this utility, the system administrator would have to set up a cron job (a job run by the `crontab` command) to run the command once, and would then have to remove the cron job when the command was finished. The `at` utility can be thought of as a fire up once and forget utility. After the job is done, the `at` command forgets about the job and must be reprogrammed to run a similar job again.

A simple way to demonstrate the `at` and `crontab` command is to have these commands execute the following command:

```
echo "message sent on "`date`" > /dev/console
```

This command will display a message like

```
message sent on May 2 11:02:13 MDT 2002
```

Because this command executes immediately and displays the current date and time, it will give the readers real time feedback as to when the `at` or `crontab` commands execute the target command.

The second part of this chapter deals with process control and is rather straightforward.

Using the `crontab` command

The `crontab` command creates what is know as a "cron job." A cron job is a command that is executed at a pre-set time and date. The `crontab` command uses a text file called a "crontab file." This file is created in the `/var/spool/cron/crontabs` directory.

The **crontab** file is named after the user that created a **cron** request. For example, if **user11** created a cron job, a text file named **user11** would be created in the **/var/spool/cron/crontabs** directory. The text file **user11** is **user11's** crontab file. This file looks like:

```
/var/spool/cron/crontabs/user11
```

This crontab file saves the date, time and command to be run. It uses the following fields:

- | | | | | | |
|------------------------------------|------------|------------|------------|---|---|
| (1) | (2) | (3) | (4) | (5) | (6) |
| (1) – Minute | | | | (0 - 59) | |
| (2) – Hour in military time | | | | (0 – 23) | |
| (3) – Day of month | | | | (1 – 31) | |
| (4) – Month of year | | | | (1 – 12) | |
| (5) – Specific weekday | | | | (0-Sunday, 1-Monday, 2-Tuesday, 3-Wednesday 4-Thursday, 5-Friday and 6-Saturday) | |
| (6) – Command | | | | | (the same thing a user would type on a command line) |

The first four fields (1) (2) (3) (4) are used to set a specific time and date for the command to run. The first two fields set the minute and hour the command is supposed to be run. The hour must be set in military 24 hour time format (3:00 p.m. = 15 , 9:00 p.m. = 21). A specific day of the week can also be set if the fifth (5) field is used. It is possible to run a command on the 5th and 15th day of the month, and also on all Mondays if fields (3) (4) and (5) are all used.

Format Rules and Examples of the Crontab Command

Some format rules for entering values are:

- Value x – a numeric value only
- Value x,y,x – use several values for a field
- value x-z – use all numbers between x and z.
- value * – use a "*" for fields to which you are not assigning a value

Table 11.1 gives some examples of **crontab** command fields. The top row shows what each field of the command refers to. The commands are shown in black on white type. Each command is followed by an explanation of that command, in black on gray type.

Minute	Hour	Day	Month	Weekday	Command
15	13,14,15	*	*	6	ufsdump 0f /dev/mnt/0 /dev/rdsk/c0t0d0s0
This command runs the backup program (ufsdump) at 15 minutes past hours 13, 14, and 15 on day 6. In other words, at 1:15 p.m., 2:15 p.m., and 3:15 p.m. on Saturday.					
23	2,4,5	*	*	1-5	ufsdump 0f /dev/mnt/0 /dev/rdsk/c0t0d0s0
This command runs ufsdump at 23 minutes past hours 2,4, and 5 on days 1-5. In other words, at 2:23 a.m., 4:23 a.m., and 5:23 a.m. on Monday through Friday.					
13	4	3	5	*	ufsdump 0f /dev/mnt/0 /dev/rdsk/c0t0d0s0
This command runs ufsdump at 13 minutes past hour 4 on day 3 of month 5. In other words, at 4:13 a.m. on May 3.					

Table 11.1 Crontab Command Fields

The `cron.allow` and `cron.deny` files

There are two special files that are used to restrict or allow the use of the `crontab` command. These files are:

```
/etc/cron.d/cron.allow
/etc/cron.d/cron.deny
```

These files are simple text files that have a user's name on each line. The rules that govern the use of these files are:

- If `cron.allow` exists, only users listed in this file can create a cron job.
- If `cron.deny` exists, any user listed here can not create a cron job.
- If both files exist, `cron.allow` is used and `cron.deny` is ignored.
- If neither file exists, only the root user can create a cron job.

The `crontab` command supports the following options:

```
crontab <options> <username>
```

<options>

`-e` Used to edit a crontab file.

`-l` Shows the user's crontab file.

`-r` Used to remove all cron jobs.

<username> This is an option for the root user only. This lets the root user work with other users' cron jobs.

Using the `crontab -r` file as the root user can be very dangerous because there are some system routines that run as the root user. Also, many software packages create cron entries as the root user, and this would destroy those entries also.

Lesson 11.1 Create a Simple Cron Job

In this lesson the reader creates a cron job and runs a simple command that displays the message `cron job run on <current date>` on a console window.

1. Log in as `user11`
2. Open a Terminal window
To open a terminal window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Tools menu item, then left click on the Terminal icon.
3. Type the command `date`
Record the time and date. They will be used later in the lesson.
4. Type the line `EDITOR=vi`
This sets the `crontab` editor. In this case the `vi` editor is being used. If the user has another command line text editor that is preferred, a different `EDITOR=<editor>` entry can be used.
5. Type the command `export EDITOR`
The `export` command is used to "export" the `EDITOR` variable. When a variable is exported it can be used and remembered in other shells owned by the same user.
6. Type the command `crontab -e`
The `crontab` command with the `-e` option opens the user's crontab file for editing.

7. Add a line *similar to* the following line to the end of the crontab file:

```
30 14 * * * echo "cron job run on" `date` >/dev/console
```

*This example sets the execution time to 2:30 p.m. (14:30 military time). Instead of this, set the execution time to five minutes past the current time. Also note that the **date** command is surrounded by grave accent marks. The grave accent marks are on the same keyboard key as the tilde.*
8. Open a Console window

To open a Console window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Hosts menu item, then left click on the Terminal Console icon.
9. Type the command **date**

Eventually a message should appear, similar to the following:

```
cron job run on May 2 2:30:00 MDT 2002
```
10. Type the command **crontab -l user11**

*This shows all the current cron jobs in **user11**'s crontab file*
11. Type the command **crontab -r user11**

*This removes all cron jobs from **user11**'s crontab file*

The at command

The **at** command is similar to the **crontab** command, except that it runs a command only once. It is also easier to set up.

The format for the **at** command is:

```
at <options> <date>
at [type the command to run]
at [press CTL + D]
```

The **at** command supports the following options:

```
at <options> <date>
```

```
<options>
```

```
-m          Send mail to the user when the job completes (provided sendmail is set up)
```

```
-t time     Time to run the command with the format options:
```

```
h          Hour: example (3 – start at 3:00 p.m. or 3:00 a.m., whatever comes next)
```

```
hh         24 hour clock (15 – start at 3:00 p.m.)
```

```
hh:mm     24 hour clock and minute (17:32 – start at 5:32 p.m.)
```

```
-r         Remove a job from at (need to have the at job number or name. You can use the atq command to to retrieve the at job name or number)
```

```
-q         Name the at job. If this is not specified, a rather cryptic at job name like 235520343 is created.
```

```
< date >
```

This option recognizes and uses the following keywords

Today

Tomorrow

Monday	Tuesday	Wednesday	
Thursday	Friday	Saturday	
Sunday			
Dec	Jan	Feb	Mar
Apr	May	Jun	Jul
Aug	Sep	Oct	Nov

Like the **cron** command, the **at** command uses two files, **at.allow** and **at.deny**. Unfortunately, these files work differently than the **cron.allow** and **cron.deny** files. Try not to get the **at** and **crontab** rules confused with each other.

The **at** command rules are:

- If the **at.allow** file exists, only users in **/etc/cron.d/at.allow** can run the **at** command.
- If the **at.deny** file exists, users in **/etc/cron.d/at.deny** can not run the **at** command.
- If both files exist, a user must be in the **/etc/cron.d/at.allow** file and the user must not be listed in **at.deny** file.
- If neither file exists, the root user is the only one that can run the **at** command.

A simple way to remember this:

- If the **at.allow** file exists and your user name is in there, you can run the **at** command.
- If the **at.deny** file exists and your user name is in there, you can not run the **at** command.
- If neither file exists, only the root user can run the **at** command.

Lesson 11.2 Using the **at** Command

In this lesson the reader creates a simple **at** job that displays the message **at command output** on a console window.

1. Log in as **user11**.
2. Open a console window
To open a console window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Hosts menu item, then left click on the Terminal Console icon.
3. Type the command **date**
Record the current time.
4. Type a command similar to **at 1430 Today** (but set the time +5 minutes in the future)
*This example given here how to set the **at** command for 2:30 or 14:30 (24 Hour Time). For this lesson, set the 24 Hour Time five minutes ahead of the current system's time.*
5. Press the Return key.
6. Type the command **echo "at command output" `date` >/dev/console**
*In this command, the word **date** is surrounded by grave accent marks. This key is above the Back Space key on a Sun keyboard.*
7. Press the CTL + D keys simultaneously.
8. Type the command **date**
9. Type the command **at -l** (at dash little L)
*The **at -l** command shows the current at jobs that are going to run.*

10. Type the command **atq**
*This command also shows future **at** jobs.
 Wait until the date used for the **at** command has passed.*
11. After the job is finished type the command **at -r myatjob**

Process Control

Sometimes, you may need to view all the processes (a program is a process) that are currently running on the system. All processes can be viewed graphically with a program called “Process manager.”

Lesson 11.3 Using the Process Manager

In this lesson the reader will start the Process Manager. The Process Manager is a very easy to use GUI tool, so this lesson is rather short.

1. Log in as the root user.
2. Open a Terminal window
To open a Terminal window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Tools menu item, then left click on the Terminal icon.
3. Type the command **/usr/dt/bin/sdtprocess &**
*The next image is a screen shot of the Process Manager with the **ID** column highlighted.*

ID	Name	Owner	CPU%	RAM	Size	Started	Parent	Command
0	sched	root	0.0	0	0	16:57:50	0	sched
1	init	root	0.0	352	1208	16:57:52	0	/etc/init -
2	pageout	root	0.0	0	0	16:57:52	0	pageout
3	fsflush	root	0.0	0	0	16:57:52	0	fsflush
53	sysevent	root	0.0	1448	2296	16:58:05	1	/usr/lib/sysevent/syseventd
62	picld	root	0.0	2168	2832	16:58:08	1	/usr/lib/picl/picld
195	skipd	root	0.0	2520	4760	16:58:16	1	/usr/sbin/skipd
201	in.ndpd	root	0.0	808	1936	16:58:17	1	/usr/lib/inet/in.ndpd
215	rpcbind	root	0.0	1328	2344	16:58:18	1	/usr/sbin/rpcbind
253	inetd	root	0.0	1784	2504	16:58:20	1	/usr/sbin/inetd -s
273	lockd	root	0.0	1480	2184	16:58:20	1	/usr/lib/nfs/lockd
274	statd	daemon	0.0	1760	2544	16:58:20	1	/usr/lib/nfs/statd
277	automoun	root	0.0	2048	3704	16:58:22	1	/usr/lib/autofs/automountd
289	syslogd	root	0.0	1744	3352	16:58:23	1	/usr/sbin/syslogd
305	cron	root	0.0	1392	2304	16:58:23	1	/usr/sbin/cron
310	nscd	root	0.0	2264	2968	16:58:24	1	/usr/sbin/nscd

4. Left Click on the **Name** column button
The running processes should be arranged in alphabetical order. Any time a column header button is left clicked, the column will be sorted by the column header.
5. In the original terminal window, type the command **admintool &**
6. Return to the Process Manager and left click on the **Name** column button until the Admintool's process is shown.
7. Highlight the Admintool process by left clicking on it only once.
8. Left click on the **Process** menu item
9. Left click on the **Kill** menu choice
The Admintool will be killed. It should disappear from the listing.
10. Double left click on the (-) symbol on the top left corner of the window.
This will close the Process Manager window and exit the program.

The `prstat` Command

The `prstat` command also shows the current system processes, but in text form. This is very valuable when a system administrator needs to see the running processes but does not have a GUI display to view the processes.

The `prstat` command supports the following options:

`prstat <options>`

- `-a` Shows processes and users at the same time..
- `-c` Continues to scroll.
- `-n` Limits the number of display lines to the number specified.
- `-p` Restricts the display to showing information about a specific process, or listed processes.
- `-s` Sorts by < cpu, pri, rss, size, time > in descending order.
- `-S` Sorts by < cpu, pri, rss, size, time > in ascending order.
- `-t` Sorts by user.
- `-u` Shows information based on a user's UID.
- `-U` Shows information based on a user's UID that is picked from a list.

Lesson 11.4 Using the `prstat` Command

In this lesson the root user executes the command `prstat -c 2 3`

1. Log in as the root user.
2. Open a Terminal window
To open a Terminal window, right click anywhere in unoccupied desktop space, left click on Tools, then left click on the Terminal icon.
3. Type the command `prstat -c 2 3`

The output of the `prstat` command follows this format:

PID – process ID number
USERNAME – who started the process
SIZE – how much memory does the process take
RSS – resident size of the process
STATE – is the process
 cpu# – running on cpu number. example `cpu0`, `cpu1`, `cpu2`
 sleep – sleeping
 run – running in the memory queue
 stop - stopped
 zombie – process died but the parent is still waiting
PRI – priority of the process (used when the CPU selects a process)
NICE – similar to the priority of a process, (used when the CPU selects a process)
TIME – cumulative CPU time (good for spotting run away processes)

CPU - % of CPU time (also good for spotting run away processes)
PROCESS/NLWD – name of process, number of LWPs (Light Weigh Process)

Key Points to Remember

It is important that a system administrator understand the **crontab** and **at** commands. There are plenty of commercial software programs, like Oracle 8i, that require the system administrator to set up a cron job for the software's internal processes. In addition to software programs, the **crontab** and **at** commands make a system administrator's life a lot easier. Imagine having to come to work every day at 10:00 p.m. just to perform a tape backup!

The Process Manager is a great tool to use if you suspect the server is running abnormally slow. The **prstat** command performs most of the same functions as the Process Manager, except it does not require a GUI display.

Chapter 12 Working with Shell Scripts

Lessons in This Chapter

Lesson 12.1 Understanding Shells.....	12-2
Lesson 12.2 Create a Simple Bourne Shell Script.....	12-4
Lesson 12.3 Using Variables in a Script.....	12-7
Lesson 12.4 Advanced Variables in the Bourne Shell Script.....	12-8
Lesson 12.5 Understanding the <code>export</code> Command.....	12-9
Lesson 12.6 Reading Values into a Script.....	12-10
Lesson 12.7 Using the <code>read</code> Command to Pause a Script.....	12-10
Lesson 12.8 Reading Multiple Values into a Script.....	12-11
Lesson 12.9 Using Command Line Arguments.....	12-12
Lesson 12.10 Using Redirection.....	12-13
Lesson 12.11 Using Pipes.....	12-14
Lesson 12.12 Using Math Expressions in the Bourne Shell Script.....	12-14
Lesson 12.13 Using Functions in a Bourn Shell Script.....	12-15
Lesson 12.14 Using Variables with Functions.....	12-16
Lesson 12.15 A Very Complex Set of Scripts.....	12-17
Lesson 12.16 Using the <code>if</code> Command.....	12-19
Lesson 12.17 Using the <code>else</code> Command.....	12-20
Lesson 12.18 Creating an <code>if/else if</code> Tree.....	12-21
Lesson 12.19 <code>if</code> Statement Testing a Number For a Range of Values.....	12-21
Lesson 12.20 The Yes Tester.....	12-22
Lesson 12.21 Using the <code>case</code> Command.....	12-23
Lesson 12.22 Loosening Up the <code>case</code> Command.....	12-23
Lesson 12.23 Creating a Menu in the Bourne Shell Script.....	12-25
Lesson 12.24 Using the <code>for</code> command.....	12-26
Lesson 12.25 Using the <code>for</code> Command With a Repetitive Command.....	12-26
Lesson 12.26 Using the <code>while</code> Command.....	12-28
Lesson 12.27 Using the <code>while</code> Command to Creating a List of Numbers.....	12-29
Lesson 12.28 Using the <code>until</code> Command to Generate Numbers.....	12-30
Lesson 12.29 Using the <code>break</code> Command.....	12-31
Lesson 12.30 Using the <code>continue</code> Command.....	12-31
Lesson 12.31 Using <code>sh -v</code> and <code>set -x</code> and <code>set +x</code>	12-32
Lesson 12.32 Create a Custom Utility.....	12-33
Lesson 12.33 Create a Run Control Script.....	12-34
Lesson 12.34 Working With the <code>.profile</code> File.....	12-35
Lesson 12.35 Capturing a Key.....	12-35

Introduction

Shell scripts can be thought of as miniature programs that are written with a text editor and kept in text files. Shell scripts work just like batch programs in MS-DOS. There are three primary types of shells that are used with Solaris: the Bourne Shell, The C Shell and the Korn Shell. A shell is a command interpreter that is a direct

interface between the user and the Kernel. With several shells, the user has the ability to choose the type of command interpreter that is used.

The shell's main responsibility is to parse the command line and handle redirection, pipes, job control and wild cards. In layman's terms, the shell watches what the user types and then translates those commands into the binary language that the Kernel can understand. If a series of commands is typed again and again, it is a good idea to put those commands in a shell script. When the script is run, the commands in it are executed. The shell script "types" the commands, instead of the user.

This chapter is going to focus on the Bourne shell script. The Bourne shell is the default shell used by the root user. Almost all of the system shells are written in Bourne. Software developers also like to use Bourne shells with their programs so that the end user can modify the script (unlike compiled software code).

Lesson 12.1 Understanding Shells

In this lesson readers will open and close various shells. The **ps** command will be used to display the currently running shells.

1. Log in as user11 or log in to a non-root account.
The user account user11 was created in Chapter 2. If this account does not exist on this system, log in as the root user, open a Terminal window and create this user with the following commands:
useradd -m -d /usr/user11 user11
mv /usr/user11/local.profile /usr/user11/.profile
passwd user11
2. Open a Terminal window.
To open a terminal window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Tools menu item, then left click on the Terminal icon.
3. Type the command **ps**
*The **ps** command shows the active processes. One of the displayed processes should be **sh**. This is the Bourne shell interpreter.*
4. Type the command **sh**
*The command **sh** starts a new instance of the Bourne shell.*
5. Type the command **ps**
There should now be two Bourne shells running. The last shell listed is the active shell. If the user quits the active shell, the previous shell will become the active shell. If the user quits all the shells, the terminal window will close.
6. Type the command **exit**
*The **exit** command takes the user out of the latest opened Bourne shell.*
7. Type the command **ps**
As can be seen, there is only one Bourne shell running now.
8. Type the command **ksh**
*The **ksh** command starts the Korn Shell.*
9. Type the command **ps**
*One of the processes that is running should be **ksh**, the Korn shell.*
10. Type the command **exit**
*The **exit** command leaves the Korn Shell*
11. Type the command **cs**
*The **cs** is the C shell interpreter. The command **cs** starts the C shell. The C shell has the prompt **<Hostname%>**.*
12. Type the command **ps**
*One of the processes that is running should be **cs**, the C-shell interpreter.*
13. Type the command **exit**
*The **exit** command leaves the C shell.*

14. Type the command **exit**

*The last **exit** should close the last Bourne shell. When that happens, the terminal window dies because the last shell interpreter has been closed.*

Bourne Shell

The Bourne Shell is the shell that is primarily used for administration on the system. Solaris 9 uses Bourne shell scripts in what are known as Run Control Scripts. These Run Control Scripts are located in the **/etc/rc0.d**, **/etc/rc1.d**, **/etc/rc2.d**, **/etc/rc3.d** and **rcS.d** directories.

When the system starts it executes the Bourne shell scripts located in these directories. These shell scripts start with a capital **S** or a capital **K**. Any shell script that does not start with a capital **S** or a capital **K** will be ignored. The scripts that start with a capital **K** run first. These scripts are used to kill a process or end a program. After all the **K** scripts (common shorthand lingo for kill scripts or scripts that start with a **K**) have been called, the **S** scripts are called. The **S** scripts are used to start a process or a program. Chapter 6 describes the boot process in detail and how the boot process uses Run Control Scripts.

If something goes wrong at startup and a script or a command called by a script hangs, the system can not start properly. The system administrator must be able to modify these shell scripts to fix the system.

As noted above, scripts are text files. A text file can have three types of permission: read, write and execute. These permissions can be seen when the command **ls -l <filename>** (that is **ls -l** little **L** **<filename>**) is typed. A shell script must be an executable file. If a file does not have the execute permission set, the file will not be run as a script.

The command **chmod u+x <filename>** is used to set the execution bit on a file. After the **chmod** command is used on a file, the command **ls -l <filename>** should show an **x** in the file's permissions. The **x** indicates that the file is executable.

If a script needs to be run in the Bourne shell, the line **#!/bin/sh** must be the first line of the script.

Quick Tip

If an error message appears saying:

```
myscript: not found      myscript = (name of a script)
```

this indicates that Solaris does not know the location of the script in the directory structure. If this happens for the script listed above, experiment with the following command structures:

```
./myscript  
or  
/directory1/directory2/myscript.  
or even  
myscript
```

Find the command structure that make Solaris execute the script. After that, use the command structure that works with your system. Understand that the **PATH** variable can be different for different users. This affects how scripts can be called to execute.

The user's **PATH** variable can have the entry **..** (a colon followed by a period). This lets the user type a command in the current working directory without the need for the **./** or **/directory/directory/script** command structure. For example, if a user had the following **PATH** variable:

```
PATH=/usr/bin:/mydata/directory32:..
```

the user could change to the script's directory and type **myscript** to make the script execute.

To temporarily change the **PATH** variable, type the commands

```
PATH=$PATH:..  
export PATH
```

Have an experienced Solaris system administrator modify your **.profile** file in your home directory to make these changes permanent. The **.profile** file is a very critical file for a user, and a mistake in this file could cause your system account to become unstable.

Lesson 12.2 Create a Simple Bourne Shell Script

In this lesson readers create a simple shell script that prints the message:

```
this is my script
```

on the screen.

In this lesson, a text editor is used to create a shell script. Then, the command **chmod u+x** is used to make it executable. Finally, the command **ls -l** is used several times to check the file's permissions.

1. Log in as **user11** or as a normal user (if not already logged in).
2. Open a Terminal window.
3. Open the Text Editor.
To open the Text Editor, right click anywhere in unoccupied desktop space. The Workspace menu will appear at the click point. Left click on the Applications menu choice, then left click on the Text Editor icon.
4. The reader can either use the Text Editor or the **vi** editor to create the following script.:

```
#!/bin/sh
echo "this is my script"
```

5. Save the file with name **myscript2**
*If the reader is unfamiliar with the **vi** editor, Chapter 4 describes how to use this program. To save a text file with the text editor, left click on **File**, then left click on **Save As...** . Make sure that you save the file to the same directory that the Terminal is in.*

6. Type the command **ls -l** (little **L**)

The output of the command should be:

```
-rw-r--r-- 1 user11 staff 35 Sep 22 20:54 myscript2
```

*The command **ls -l** shows the permissions on a file. In this case the first character (**-**) indicates the file is a regular file. The next three characters (**-rw**) show that the user has read/write permission. The next three characters (**r--**) show that others in the same group have read permission, and the last 3 characters (**r--**) show that others have read only permission.*

7. Type the command shown below

```
myscript2
```

*This command tells Solaris to execute the script **myscript2**. However, at the moment, this command will not work! It is designed to show some typical error messages that Solaris 9 shows with scripts.*

One error message that you might see is the following:

```
./myscript2: not found
```

This indicates that Solaris does not know the location of the script in the directory structure.

If this happens, experiment with the following command structures:

```
./myscript2
```

or

```
/usr/user11/myscript2
```

or even

```
myscript2
```

Find the command structure that make Solaris recognize the script's location.

Another error message that should appear is something like:

```
myscript cannot execute
```

*The script should not be able to run, because it has not yet been made executable. As shown in step 9, the **chmod** command must be run to make the script executable.*

8. Type the command **chmod u+x myscript**
*The command **chmod u+x** sets the execution bit **x** on the file **myscript** and makes the file executable. The **u+x** indicates that the execution bit is being set for the user (**u**).*
9. Type the command **./myscript2**
*Now that it has been made into an executable file, the script named **myscript2** should work.*
10. Type the command **ls -l**
The output of the command should be:

```
-rwxr--r-- 1 user11 staff 35 Sep 22 20:54 myscript2
```

11. Notice that the **myscript2** file now has an **x** as the fourth character. As this shows, because the file is now executable, the user has execute permission.
12. Do not close the Terminal window or the Text Editor.

The Terminal window and Text Editor will be used in some other examples.

The **chmod** command can be used with either numbers (**1-7**) or letters (**r w x**) to set permissions. As shown above, for a text file to be used as a script, its status must be set to executable. You can use any of the methods shown below.

The **chmod** command can use the following numbers to set permissions:

- 7 - Read/write/execute**
- 6 - Read/write**
- 5 - Read/execute**
- 4 - Read**
- 3 - Write/execute**
- 2 - Write**
- 1 - Execute**

By specifying three numbers with the **chmod** command, the permissions for a file can be set for the user, for other people in the user's group and for all other users (in that order). Looking at the list of numbers above, we can see the results of the following commands:

Command	Permissions Set By the Command	Display of Permissions
chmod 744	user rwx group r others r	rwx r-- r--
chmod 317	user -wx group --x others rwx	-wx --x rwx
chmod 215	user -w- group --x others r-x	-w- --x r-x
chmod 777	user rwx group rwx others rwx	rwxrwxrwx

Instead of using numbers, the letters **r w x** and **u g o** can be used to set permissions. With this method, the letters **r**=read, **w**=write, **x**=execute, **u**=user, **g**=group, and **o**=others are used to set permissions.

Here are some examples:

chmod u=x Sets the permissions u = --x g= not affected , o = not affected .The group and others permissions are not changed by this command.

chmod u+x Adds the permission u=x to whatever the permissions were before.

chmod u-x Removes the permission u=x from whatever the permission were before.

chmod u=rw,g=rwx Sets the permissions u= rw- g= rwx o= not affected.

chmod o-rx Removes the read and execute permissions from others, u = not affected, g = not affected.

chmod u=rwx,g=rwx,o=rwx
Sets the permissions u=rwx g=rwx o=rwx on a file

Using Variables

Variables are used to save a value for future use. The easiest variable to use is one that saves characters. With this type of variable, a single character or text message can be saved. A variable is saved with the script line **<variable>=<value>**. For example: **var1=cat** or **var2=mice**.

The Solaris 9 operating system creates some variables automatically. These variables can be seen by typing the command **env**. The variables are known as Keyword Shell Variables. They can be changed by a user if needed.

The **echo** command can be used to display the contents a variable. The **echo** command ordinarily only displays whatever follows it on the screen. To display the contents of a variable, type the name of the variable with a dollar sign in front of it. For example, for a variable called **myvar** the command **echo \$myvar** would display the contents of that variable.

Lesson 12.3 Using Variables in a Script

In this lesson users create some variables and then display those variables on the screen with the **echo \$VARIABLE** command.

1. Log in as user11 or an ordinary user.
2. Open the Text Editor.
To open the Text Editor, right click anywhere in unoccupied desktop space. The Workspace menu will appear at the click point. Left click on the Applications menu choice, then left click on the Text Editor icon.
3. Open a Console window.
In previous lessons the reader was instructed to open a Terminal window. In this lesson we will open a Console window. Understand that both a Console window and a Terminal window can be used to enter text on the screen. To open a Console window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Hosts menu item, then left click on the Terminal Console icon.
4. Type the following three lines into the Text Editor:

```
#!/bin/sh
mydog=spot
echo "my dog is named $mydog"
```
5. Save the script with the name **myscript3**
To save a text file with the Text Editor, left click on File, then left click on Save As...
6. In the Console window, type the command **chmod 777 myscript3**
*The command **chmod 777** sets rwxrwxrwx permissions on the file.*
7. Type the command **myscript3**
The output of the script should be:

```
my dog is named spot
```

If an error message appears saying:

```
myscript3: not found
```

this indicates that Solaris does not know the location of the script in the directory structure. Experiment with the following command structures:

```
./myscript3
or
/usr/user11/myscript3
or even
myscript3
```

Find the command structure that make Solaris recognize the script's location.

8. Type the command **env**
*The **env** command shows the Keyword Shell Variables.*
9. Modify **myscript3** so it looks like this:

```
#!/bin/sh
mydog=spot
```

```

echo "my dog is named $mydog"
echo "the HOME system variable is $HOME"
echo "the TERM system variable is $TERM"
echo "the MANPATH system variable is $MANPATH"
echo "the LANG system variable is $LANG"
echo "the DISPLAY system variable is $DISPLAY"
echo "the LOGNAME system variable is $LOGNAME"

```

To save a text file with the Text Editor, left click on File, then left click on Save As...

10. Save the file as **myscript3.2**.

11. Type the command **./myscript3.2**

The output of the command should be something like:

```

my dog is named spot
the HOME system variable is /usr/user11
the TERM system variable is dtterm
the MANPATH system variable is /usr/bin::usr/dt/bin:usr/openwin/bin:bin:usr/ucb
the LANG system variable is en_US.ISO8859-15
the DISPLAY system variable is :0.0
the LOGNAME system variable is user11

```

Different systems are set up differently, so your display may be different.

To clear the value of a variable add a line such as "**myvariable=**" to the script. This redefines the variable as nothing. The command **unset** can also be used to clear a variable. For example, the line **unset myvariable** would also clear out the value of the variable.

To save more than one word in a variable, put quotation marks around all the words, as in the second example below.

```

myvariable = how are you          - This command will not work
myvariable="how are you"         - This sets the variable to "how are you"

```

A variable can be made read-only with the command **readonly <variable name>**. Read-only variables can not be changed later in the script. In the Bourne shell, a variable is ordinarily local only to the process that created the variable. Another process or program can not use the variable from your script unless you type the command **export myvariable**.

Lesson 12.4 Advanced Variables in the Bourne Shell Script

In this lesson readers will work with some of the advanced features of variables in the Bourne shell. A variable will be created with a string of text. The variable will be cleared with the **unset** command. Later, the variable will be protected with the **readonly** command so that it can not be changed later in the script.

1. Log in as **user11** or an ordinary user.
2. Open the Text Editor.
3. Open a Console window.
4. Create or copy the following lines into a script:

```

#!/bin/sh
myvar1="how are you"
echo "The value of myvar1 is - $myvar1 - before the unset command."
unset myvar1
echo "The value of myvar1 is - $myvar1 - after the unset command."
myvar1="new myvar1 value"
echo "the value of myvar1 is - $myvar1 - as defined again"
readonly myvar1
unset myvar1

```

- echo "the value of myvar1 is now - \$myvar1"**
- Save the text file with the name **myscript4**
 - Type the command **chmod 777 myscript4**
*The command **chmod 777** gives read/write/execute permission to **myscript4***
 - Type the command **./myscript4**
*The script will stop with an error, because it attempts to use the **unset** command with the readonly variable **myvar1**. You will see the error message:
./myscript4 myvar1: is read only*
 - Modify the ninth line of the script to read as follows:
unset myvar1
Notice that the # character has been added at the start of the line. This converts the line to a non-executable line, or comment. When the script is run, this line will not be executed. This means that the Bourne Shell will not try to change the value of myvar1.
 - Execute the script with the command **./myscript4**
*The script should work this time, now that it does not try to change the value of a **readonly** variable.*

Lesson 12.5 Understanding the export Command

In this lesson readers will work with two shell scripts. In the first script, the variables **myvar1** and **myvar2** are defined. In **script1** the variable **myvar1** is not exported, while the variable **myvar2** is exported. In the second script, only the **myvar1** variable is exported. Also in this lesson, readers will learn how one script can call another script

- Log in as **user11** or an ordinary user.
- Open the Text Editor.
- Open a Console window.
- Create the following script:
#!/bin/sh
myvar1="myvar 1 this is not exported"
myvar2="myvar 2 this is exported"
export myvar2
echo "script 1:the value of myvar1 is : \$myvar1"
echo "script 1:the value of myvar2 is : \$myvar2"
./script2
*The first line **#!/bin/sh** calls the Bourne shell. The next two lines define **myvar1** and **myvar2** with a text string. The fourth line (**export myvar2**) exports the variable **myvar2**. The next two lines display the values of **myvar1** and **myvar2**. The last line (**./script2**) executes the second script, which will be created below.*
- Save this script as **script1**
- Type the command:**chmod 777 script1**
*The command **chmod 777** gives read/write/execute permission to **script1**. (Optional: use the command **ls -l** to view the permissions).*
- Create the following script:
#!/bin/sh
echo "script 2: the value of myvar1 is \$myvar1"
echo "script 2: the value of myvar2 is \$myvar2"

*The first line of the script (**#!/bin/sh**) calls the Bourne shell. The next two lines display the value of **myvar1** and **myvar2**.*
- Save this script as **script2**
- Type the command **chmod u+x script2**
*The command **chmod u+x** makes **script2** executable. You could also use the command **chmod 777 script2***
- Type the command **./script1**

Look at the results carefully. Notice that within **script1**, both **myvar1** and **myvar2** have values, but that in **script2**, only the exported variable **myvar2** has a value.

Reading Values into a Script

Up to this point, all variables have been defined within the script. Now it is time to let the user type in a response to a question and have that response saved in the script. The values are recorded by the **read** command. The **read** command takes text from the keyboard and saves it in a variable. When the user presses the Return key, the value is read into the **read** command.

Lesson 12.6 Reading Values into a Script

This lesson introduces the readers to the read command. Here, the read command is used to record the user's first name in a variable. The variable is then printed with the echo command.

1. Log in as **user11** or as an ordinary user.
2. Open the Text Editor.
3. Open a Console window.
4. Type the following script:

```
#!/bin/sh
echo "what is your name?"
read myname
echo "your name is $myname"
```
5. Save the script as **myscript6**
6. Type a **chmod** command that will make the script executable. Look at previous lessons if necessary.
7. Type the command **./myscript6**
*When **myscript6** runs, it simply asks the user for his or her name and saves this value in the variable named **myname**. The last line of the script uses the **myname** variable to print out the user's name.*
8. To make the script more visually appealing, replace the line

```
echo "what is your name?"
```

with the line

```
echo "what is your name? \c"
```

*The **\c** option makes the cursor stay on the same line as the prompt, rather than moving to the start of a new line. The user answers the question on the same line where it appears.*

Lesson 12.7 Using the read Command to Pause a Script

The **read** command can also be used to pause a script until the user presses the Return key. In this example, the **read** command is used to read a value into a variable called **ignore**, but the value of that variable is never used. This script also shows the reader how to open a console and how to send a message to the console window.

1. Log in as **user11** or as an ordinary user
2. Open the Text Editor.
3. Open a Terminal window.
4. Type the following script in the Text Editor:

```
echo "line executed before the read command"
echo "press the Return key to continue:"
read ignore
echo "line executed after the read command"
```
5. Save the script as **myscript7**
6. Type the command **chmod 777 myscript7** in the Terminal window.
7. Type the command **./myscript7**

This script should create the following output:

```
line executed before the read command
press the Return key to continue
after the Return key is pressed
line executed after the read command
```

Lesson 12.8 Reading Multiple Values into a Script

The **read** command can be used to read in more than one variable at a time. In this lesson the **read** command will be used to capture a person's full name (first, middle, last).

1. Log in as **user11** or as an ordinary user
2. Open the Text Editor.
3. Open a Terminal window.
4. Type the following script:

```
#!/bin/sh
echo "what are your first, middle, and last names? \c"
read first middle last
echo "Your first name is $first"
echo "Your middle name is $middle"
echo "Your last name is $last"
```

Most of the script is easy to understand. The \c option makes the cursor stay on the same line as the prompt, rather than moving to the start of a new line. This produces the display:

What are your first, middle and last names? <prompt is still on the same line, which looks better >

5. Save the script as **myscript8**
6. Type the command **chmod 777 myscript8**
7. Type the command **./myscript8**

Assume that the user typed "Karen Smith Anderson" at the prompt and then pressed the Return key. The prompt and its answer will appear as:

*What are your first, middle, and last names? Karen Smith Anderson
The output is :*

```
Your first name is Karen
Your middle name is Smith
Your last name is Anderson
```

Command Line Arguments

Instead of getting information by prompting a user to respond to a series of questions, some shell scripts use *arguments* that appear after the shell script's name to get information. For example, a script could be created with the name **mydiskinfo**. It could be designed to let the user type the command **mydiskinfo floppy** to tell the script that he or she wanted information on the floppy drive, or **mydiskinfo harddrive** to obtain information on the hard drive.

This script uses **floppy** and **harddrive** as its arguments. To record the first word typed after the script name, a script uses the reserved variable **\$1** for its first argument. To record the second word typed after the script name, a script uses the reserved variable **\$2** for the second argument. The reserved variable **\$0** is the name of the script itself.

Lesson 12.9 Using Command Line Arguments

This lesson uses a simple script that takes the first argument (a person's name) and sets the **myname** variable to the first argument. The echo command is used to display the text

your name is < your real name >

1. Log in as **user11** or as an ordinary user
2. Open the Text Editor.
3. Open a Terminal window.
4. Create and save the following script, named **myscript9**

```
#!/bin/sh
myname=$1
echo "your name is $myname"
```
5. Type the command **chmod 777 myscript9**
6. Run the script, using your name as the first argument. If your name is Karen, type **./myscript9 Karen** < type in your name instead of Karen >

*The output of the script should be: **Your name is Karen** < or your name >*

7. Modify the script so it looks as follows. Then save the script as **myname2**.

```
#!/bin/sh
nameofscript=$0
myname=$1
mylastname=$2
echo "Your full name is $myname $mylastname"
echo "The name of the script is $nameofscript"
```

*The line **mylastname=\$2** takes the second argument or \$2. The line **nameofscript=\$0** takes the "#0" argument, the name of the script.*

8. Type the command **./myname2 Karen Smith** < type in your first and last name instead of Karen Smith >
The output of the script should be:

```
Your full name is Karen Smith
The name of the script is myname2
```

Saving Output To a Text File

Scenario: A system administrator creates a wonderful script that reports on all the files that have been modified in the last three days. The script is run, and over 500 file names fly past on the screen. Even if it were possible to grab a pencil and paper and write down all the file names by hand, consider the consequences. Say that the system administrator gives such a handwritten list to the team lead. In that case, it might be time for the system administrator to start updating his or her resume!

To solve this problem, the Bourne shell has a function called *redirection*. Instead of having the output go to the screen, the output can be sent into a text file. To use redirection, at the end of the command line, add the (>) symbol (which says to redirect the output), and a filename to put the output into. For example, the command **ls -l > mylisting** puts the output of the **ls -l** command into a file called **mylisting**. When output is redirected into a file, it does not appear on the screen.

Technically, redirection works by taking the STDOUT (Standard OUT) that would go to the monitor and redirecting it to a text file. The greater than symbol (>) is used to create a text file and to add the STDOUT to the text.

You can also use the symbol (`>>`) to redirect output. In this case, if the file already exists, the output will be *appended* to the end of it. If the file does not exist, it will be created, just as if the (`>`) symbol had been used. For example, the command `ls -l >> mylisting` appends the output of the `ls -l` command to the current contents of an existing file called `mylisting`.

You can also use the symbol (`>>`) to redirect output. In this case, if the file already exists, the output will be *appended* to the end of it. If the file does not exist, it will be created, just as if the (`>`) symbol had been used. For example, the command `ls -l >> mylisting` appends the output of the `ls -l` command to the current contents of an existing file called `mylisting`.

WARNING

- When text is redirected with the `>` operator into an *existing* file, the file's contents will be permanently *overwritten* by the new text. Do not use the `>` operator on an existing file that you want to keep!
- You can safely use the `>>` operator to append text to the end of an existing file.

There is one other redirection operator. If (`<`) is used with a command, information is taken from a text file, rather than from the keyboard, and is then sent to a command.

Lesson 12.10 Using Redirection

In this lesson readers will use the (`>`), (`<`) and (`>>`) redirection symbols. Remember, the (`>`) symbol is used to create and add text to a file. The (`>>`) symbol adds extra text to an existing file without damaging the original text contents. The (`<`) symbol extracts file from a text file.

1. Log in as `user11` or as an ordinary user.
2. Open the Text Editor.
3. Open a Terminal window.
4. Type the command `date` (in the terminal window).
The system date should now appear on the screen.
5. Type the command `date > datefile`
The system date is now written to a text file named `datefile`
6. Type the command `cat datefile`
The `cat` command shows the contents of the `datefile` text file.
7. Create the following script:

```
#!/bin/sh
ls /etc > etcfile
```

The command uses the `ls` command to list all the files in the `/etc` directory. The output is redirected to the text file `etcfile`.
8. Save the script as `myscript10`
9. Type the command `chmod 777 myscript10`
10. Type the command `./myscript10`
The `myscript10` file creates a new text file with the name `etcfile`.
11. Type the command `cat etcfile`
The `etcfile` file contains the system date.
12. Type the command `wc -l < etcfile`

The **wc** (word count) command counts the words in its input. Normally, the input comes from the keyboard. Here, the **<** operator tells **wc** to read the input from **etcfile** instead. The screen now shows how many words are in **etcfile**.

13. Type the command **echo "add line 1" > myfile1**

The file myfile1 is created. It has one line: "add line 1"

14. Type the command **echo "add line 2" >> myfile1**

The line "add line 2" is appended to the end of the file myfile1.

15. Type the command **cat myfile1**

As shown onscreen, the echo commands created myfile1 and added a second line to it, using > and >> for redirection.

Lesson 12.11 Using Pipes

The standard output from one command can be *piped* directly into the standard input of another command. For example, take the command

```
ls /etc | grep "^a"
```

This command takes the output of the command **ls /etc** and instead of putting this output on the screen (the screen is STDOUT), it pipes the information directly into the **grep** command. The **grep** command here takes the standard input (STDIN) and searches for all words that start with the letter **a**. The pipe symbol (**|**) is created by pressing the key that is located above the Return key on a PC keyboard. The pipe symbol is above the Back Space key on a Sun keyboard.

1. Login as **user11** or as an ordinary user.

2. Open a Terminal window.

3. Type the command **ls /etc > etcfile**

The command uses the ls command to list all the files in the /etc directory. The output is redirected to the text file etcfile.

4. Type the command **cat etcfile | grep "^a"**

The command cat etcfile | pipes the output of the cat etcfile command into the grep command. The grep command is a pattern matching command. The command grep "^a" finds all the files and directories in the /etc directory that start with the letter a.

Mathematical Expressions in the Bourne Shell Script

It is possible to do some basic mathematics in the Bourne shell script. This includes addition, subtraction, multiplication and division. Unfortunately the Bourne shell can only work with integers, so if the actual answer is something like 2.35 or 4.362, it will be reported (incorrectly) as a whole number such as 2 or 4. If you need to work with math that produces decimal numbers, use a more advanced shell, like the C shell.

Lesson 12.12 Using Math Expressions in the Bourne Shell Script

This lesson will demonstrate how to use simple math operations, such as addition, subtraction, multiplication and division in a shell script.

1. Log in as **user11** or as an ordinary user

2. Open the Text Editor.

3. Open a Terminal window.

4. Create the following script

```
#!/bin/sh
a=10
b=4
```

```

addval=`expr $a + $b`
subval=`expr $a - $b`
mulval=`expr $a \* $b`
divval=`expr $a / $b`
echo “the value of $a + $b = $addval”
echo “the value of $a - $b = $subval”
echo “the value of $a * $b = $mulval”
echo “the value of $a / $b = $divval”

```

*IMPORTANT: Note that the lines that perform math (the ones with the variables **addval**, **subval**, **mulval** and **divval**) use the **grave accent character** (```) around the mathematical expressions. **This is not the single quote character** (`'`).*

On a Sun keyboard, the grave accent character is the key just above the Back Space Key. On a PC keyboard, this key is usually above the Tab key on the left side of the keyboard, on the same key as the tilde (~) key.

5. Save the script with the name **myscript12**
6. Type the command **chmod 777 myscript12**
7. Type the command **./myscript12**

The Bourne shell script numbers should create the following output:

```

the value of 10 + 4 = 14
the value of 10 - 4 = 6
the value of 10 * 4 = 40
echo “the value of 10 / 4 = 2

```

*Notice that on the last line, **10 / 4 = 2** is not correct. Obviously, it should be **10 / 4 = 2.5**. Unfortunately, the Bourne shell only works with integers. If you want to work with floating point numbers and other detailed mathematics, try using another shell.*

Using Functions in a Bourne Script

The Bourne shell script supports functions. Functions let you create mini-programs within a script. Once a function is established, a script can call that function, rather than having to include all the operations that the function carries out. This is good because it helps to clean up a shell script. This makes the script easier to read and understand.

Lesson 12.13 Using Functions in a Bourn Shell Script

The reader will create a simple function (adding two numbers) inside a Bourne shell script and will then have the script call that function.

1. Log in as **user11** or as an ordinary user
2. Open the Text Editor.
3. Open a Terminal window.
4. Create the following script:

```

#!/bin/sh
# Start of the add_numbers( ) function
add_numbers (
{
echo “value of $1 is $1”
echo “value of $2 is $2”
c=`expr $1 + $2`
}
# End of the add_numbers ( ) function
add_numbers 5 1
echo “the value of $1 + $2 = $c “

```

The script works as follows:

- a. In the first line, the expression `#!/bin/sh` invokes the Bourne shell script.
 - b. The other lines that start with the `#` character are comments.
 - c. The lines between the comments are the function `add_numbers ()`.
 - The starting curly bracket `{` marks the start of the function.
 - The line `echo "value of $1 is $1"` displays the value of `$1`.
 - The line `echo "value of $2 is $2"` displays the value of `$2`.
 - The variables `$1` and `$2` in this case take the first and second arguments that are passed to the function when it is run. As shown in the line after the second comment, in this case `$1 = 5` and `$2 = 1`.
 - The line `c=`expr $1 + $2`` evaluates `$1 + $2` and puts the result in the variable `c`.
 - The ending curly bracket `}` marks the end of the function.
 - d. The line `add_numbers 5 1` calls the function `add_numbers` with the first argument `5` and the second argument `1`.
 - e. The final line shows what was added, and the final result.
5. Save the script with the name `myscript13`.
 6. Type the command `chmod 777 myscript13`.
 7. Type the command `./myscript13`.

The output should be
the value of `5 + 1 = 6`

Lesson 12.14 Using Variables with Functions

In this lesson, the user creates a slightly more complex script that uses the `read` command to read in numbers from the command line. These numbers are then added in the `add_numbers ()` subroutine.

1. Login as user `l1` or an ordinary user.
2. Open the Text Editor.
3. Open a Terminal window.
4. Call up `myscript13` and find the line that says `# End of the add_numbers () function`. Delete all the lines after that line, then add the following lines at the end:

```
echo "type in the first number : \c"  
read a  
echo "type in the second number: \c"  
read b  
add_numbers $a $b  
echo "the value of $a+ $b = $c "
```

The finished script should look like this:

```
#!/bin/sh  
# Start of the add_numbers() function  
add_numbers ()  
{  
echo "value of $1 is $1"  
echo "value of $2 is $2"  
c=`expr $1 + $2`  
}  
# End of the add_numbers () function  
echo "type in the first number : \c"  
read a  
echo "type in the second number: \c"  
read b
```

- ```

add_numbers $a $b
echo "the value of $a+ $b = $c "

```
5. Save the script with the name **myscript14**
  6. Type the command **chmod 777 myscript14**
  7. Type the command **./myscript14**  
*The output of the command should be something like:*

```

Type in the first number : 23
Type in the second number: 85
the value of 23 + 85 = 108

```

## Lesson 12.15 A Very Complex Set of Scripts

This lesson is used as a review of the previous material. Two scripts are created that, combined with each other, add two numbers together.

1. Log in as **user11** or as an ordinary user
2. Open the Text Editor.
3. Open a Terminal window.
4. Create the following script:

```

#!/bin/sh
echo "type in the first number : \c"
read num1
echo "type in the second number : \c"
read num2
./complex2 $num1 $num2

```
5. Save the script with the name **complex1**
6. Create the following script

```

#!/bin/sh
a=$1
b=$2
add_numbers (
{
num3=`expr $1 + $2`
echo "value of $a + $b = $num3"
}
add_numbers $a $b

```
7. Save the script with the name **complex2**
8. Type the command **chmod 777 complex1**
9. Type the command **chmod 777 complex2**
10. Type the command **./complex1 3 7**

*These scripts do the following things:*

1. The **complex1** script reads the two variables **num1** and **num2** when the user types them.
2. In the last line of **complex1** these variables are used as the arguments for the **complex2** script: **complex2 num1 num2**
3. The **complex2** script takes the two arguments and assigns them to the variables **\$a** and **\$b**
4. The function **add\_numbers** is defined.
5. When it is called in the last line of the script, **add\_numbers** adds the numbers and displays the output in the statement  
**echo "value of \$a and \$b = \$num3"**

## Control Statements

Most UNIX administrators are familiar with control statements. The Bourne shell supports the following control statements:

### The **if** control statement

The **if** control statement is used to test a conditional statement. Figure 12.1 and Figure 12.2 show the construction of the **if** statement.

```
if (test expression)
 then
 commands
fi
```

**Figure 12.1 The **if** Control Statement**

If the test expression is **true** all statements between the **then** and **fi** are executed. If the test expression is **false**, nothing is done. The statements between the **then** and **fi** are ignored, and program execution continues with the next line.

The **else** statement can be used to specify what happens if the control statement is **false**:

```
if (test expression)
 then
 commands
 else
 commands
fi
```

**Figure 12.2 The **if...then...else** Control Statement**

If the test expression is **true** all statements between the **then** and **else** are executed. If the test expression is **false** all statements between the **else** and **fi** are executed. The command **elseif** can be used instead of **else** to test a new set of expressions.

The test expression is any statement that is either true or false. For example, if a variable **\$dog** had the value **Niki** and the **if** statement had the test expression (**Nikki = \$dog**) the expression would be considered **true** and the commands following **then** would be executed. But if the variable **\$dog** were changed to **zeppi**, the test expression (**Nikki = \$dog**) would be **false** and the commands following the **else** would be run, instead.

Valid test expressions are:

### Integer Tests

|                                    |                                            |
|------------------------------------|--------------------------------------------|
| <code>integer1 -eq integer2</code> | integer1 is equal to integer2              |
| <code>integer1 -ne integer2</code> | integer1 is not equal to integer2          |
| <code>integer1 -gt integer2</code> | integer1 is greater than integer2          |
| <code>integer1 -ge integer2</code> | integer1 is greater or equal to integer2   |
| <code>integer1 -lt integer2</code> | integer1 is less than integer2             |
| <code>integer1 -le integer2</code> | integer1 is less than or equal to integer2 |

## Text Tests

|                |                                                                       |
|----------------|-----------------------------------------------------------------------|
| text1 = text2  | Value of text1 equals text2 (same characters)                         |
| text1 != text2 | Value of text 1 does not equal text 2 (different characters)          |
| text           | The text is not null (a value has been assigned to this variable)     |
| -n text        | The length of the string is not zero (variable contains some text)    |
| -z text        | The length of the string is zero (variable does not contain any text) |

## File Tests

|         |                                            |
|---------|--------------------------------------------|
| -b file | block special file                         |
| -c file | character special file                     |
| -d file | directory                                  |
| -f file | regular file                               |
| -g file | setGID present                             |
| -k file | sticky bit present                         |
| -p file | file is a named pipe                       |
| -r file | file is readable                           |
| -s file | file has some content, is not a blank file |
| -w file | file is writeable                          |
| -x file | file is executable                         |

## Lesson 12.16 Using the if Command

In this lesson readers will use the **if** statement on a test expression. In this case, the test expression is used to determine if a file exists. At first the file does not exist, so the test expression is **false** and the statement **echo "myfile23 exists"** is not run. Later, the **touch** command is used to create a file with the name **myfile23**. After the file is created, the **if** statement evaluates the test condition as **true** and the command **echo "myfile23 exists"** is run.

1. Log in as **user11** or as an ordinary user.
2. Open the Text Editor.
3. Open a Terminal window.
4. Type in the following script.

*Please note that the **if** command is very picky about things. Type the script exactly as it appears. This includes the square brackets in the second line, the blank space after the left square bracket, and the blank space before the right square bracket.*

```
#!/bin/sh
if [-f myfile23]
then
 echo "myfile23 exists"
fi
```

5. Save the script as **myscript16**
6. Type the command **chmod 777 myscript16**
7. Type the command **./myscript16**  
*When the script runs, no output is produced. This is because the test condition [ -f myfile23 ] tests to see if a file named **myfile23** exists. The file **myfile23** does not exist, so the command **echo "myfile23 exists"** is never run.*
8. Type the command **touch myfile23**  
*The command **touch <filename>** creates a blank file. In this case **touch myfile23** creates a blank file with the name **myfile23**.*
9. Type the command **./myscript16**

This time, the script should produce the output **myfile23 exists**. After **myfile23** was created by the **touch** command, the test condition `[ -f myfile ]` evaluated to **true**.

## Using the else option with the if Command

Most users don't like a script that dies without giving some kind of error message or warning message that explains what when wrong. In the previous lesson, when the script did not find the **myfile23** file, it ended without giving any type of messages.

To avoid that, the following lesson uses the **else** keyword to type an alternate message if the **if** test condition evaluates to **false**.

### Lesson 12.17 Using the else Command

In this example the **else** command creates an alternate message if the original **if** test condition is **false**. This script is nicer to users than the last one, because it tells users why this script ended the way it did.

1. Log in as **user11** or as an ordinary user.
2. Open the Text Editor.
3. Open a Terminal window.
4. Type in the following script (remember to type it exactly as shown):

```
#!/bin/sh
if [-f myfile25]
then
 echo "myfile23 exists"
else
 echo "myfile23 does not exist, ending script"
fi
```

5. Save the script with the name **myscript17**
6. Type the command **chmod 777 myscript17**
7. Type the command **rm myfile23**  
*The command **rm myfile23** deletes the **myfile23** file.*
8. Type the command **./myscript17**  
*The script should display the message **myfile23 does not exist, ending script**. This is because the test condition `[ -f myfile23 ]` is **false**. In this case, the **if** keyword will not run the commands after the word **then**. Instead, the **else** keyword will run the commands that appear after it. In other words, the **else** keyword works when the **if** condition is **false**.*
9. Type the command **touch myfile23**
10. Type the command **./myscript17**  
*This time, the script should produce the output **myfile23 exists**. This happens because **myfile23** was created by the **touch** command. and the test condition `[ -f myfile ]` is now **true**.*

## Creating an if/else if Tree

Sometimes, any of several different conditions could be **true**. When this is the case, it is helpful to use a construction technique with the **if** command that we will call an "**if/else if** tree." By using one **if** statement and multiple **else if** statements, a script can handle multiple possibilities with ease.

```
if [test condition]
then
 command(s)
```

```

else if [test condition]
then
 command(s)
else if [test condition]
then
 command(s)
fi
fi
fi

```

With an **if/else if** tree, a condition can have numerous tests performed on it. No test has any higher precedence than any other. This structure simply gives the script a chance to perform multiple tests on a variable or condition.

## Lesson 12.18 Creating an **if/else if** Tree

In this lesson a simple **if/else if** tree is created that tests the value of a variable. As we will see, the script will continue to look for a matching value if the first if statement is **false**.

1. Log in as **user11** or as an ordinary user
2. Open the Text Editor.
3. Open a Terminal window.
4. Type in the following script:

```

#!/bin/sh
echo "Type in the value of a: \c"
read a
if [$a -eq 1]
then
 echo " a = 1 "
else if [$a -eq 2]
then
 echo " a = 2 "
else if [$a -eq 3]
then
 echo " a = 3 "
fi
fi
fi

```

5. Save the script with the name **myscript18**
6. Type the command **chmod 777 myscript18**
7. Type the command **./myscript18**  
*When the script runs, enter 1, 2, or 3 as the value for a. The script will move down the **if/elseif** tree. When it finds a value that it "knows about" it will print that value. None of the if or elseif checks has greater importance to the program than any of the others.*
8. Type the command **./myscript18**  
*This time, enter the value 4. No output will be produced. Can you figure out why?*

### Other Useful **if** Test Conditions

This section features a series of lessons that demonstrate some very useful **if** statements.

## Lesson 12.19 **if** Statement Testing a Number For a Range of Values

In this lesson, the script tests a user's guess number for a range of values.

1. Log in as **user11** or as an ordinary user
2. Open the Text Editor.
3. Open a Terminal window.
4. Type in the following script:
 

```
#!/bin/sh
echo "pick a number between 10 - 30 ? \c"
read myvar1
if [`expr $myvar1` -lt 10 -o `expr $myvar1` -gt 30]
then
echo " Number not chosen between 10 - 30 "
else
echo " Number was between 10 - 30 "
fi
```
5. Save the script with the name **myscript19**
6. Type the command **chmod 777 myscript19**
7. Type the command **./myscript19**  
*The test condition uses the **expr** command to find the numeric value of the variable **myvar1**. The test conditions are **-lt** (less than), **-gt** (greater than) and **-o** (or).*

## Lesson 12.20 The Yes Tester

In this lesson, an answer is tested to see if it is **y** or **Y** answer.

1. Log in as **user11** or as an ordinary user
2. Open the Text Editor.
3. Open a Terminal window.
4. Type in the following script:
 

```
#!/bin/sh
echo "are you sure (y or Y)? \c"
read myvar1
if ["$myvar1" = Y -o "$myvar1" = y]
then
echo " The user typed yes "
fi
```
5. Save the script with the name **myscript20**
6. Type the command **chmod 777 myscript20**
7. Type the command **./myscript20**  
*The only complex part of this script that should be explained is the test condition. The variable **\$myvar1** is enclosed in double quote marks (") so that its alphanumeric value can be compared against **y** or **Y**. The **-o** option means "or". Notice that the script produces output only if the user types **y** or **Y** because there is no **else** or **else if** condition to tell it what to do otherwise.*

## The case Command

The **case** command works just like the **if/else if** tree (but is more convenient to use). It checks a variable for various conditions. When a condition is met, the statements that go with it are executed. Each tested condition begins with the condition and ends with two semicolons (**;;**).

The **case** command uses the asterisk character (**\***) as a catch-all for any other value than the tested values. The keyword **esac** (case spelled backward) is used to end the **case** command's structure.

case variable in

```

 value1)
 statements
 ;;
 value2)
 statements
 ;;
 *)
 statements
 ;;
esac

```

## Lesson 12.21 Using the **case** Command

In this lesson the **case** command is used to ask about a user's car.

1. Log in as **user11** or as an ordinary user
2. Open the Text Editor.
3. Open a Terminal window.
4. Type the following script:

```

#!/bin/sh
echo "who makes your car? \c"
read mycar
case "$mycar" in
Ford)
 echo "An American car"
 echo "Made in Detroit"
 ;;
Mercedes)
 echo "Yeah, sure you own one!"
 ;;
Hyundai)
 echo "An Asian car"
 echo "Made in Korea"
 ;;
*)
 echo "Never heard of that car"
esac

```

5. Save the script with the name **myscript21**
6. Type the command **chmod 777 myscript21**
7. Type the command **./myscript21**

Type in **Ford**, **Mercedes**, **Hyundai** or any other car company's name. Notice that any car other than the ones tested for gets the response "Never heard of that car."

## Lesson 12.22 Loosening Up the **case** Command

People have the bad habit of making typing mistakes. Instead of typing the word Ford, some people might type Foord or Fod or ford (all lower case) instead. To allow for this type of variation, the **case** command supports what are known as *regular expressions*. Most people know that the asterisk (\*) represents any character or string of characters. As shown below, there are other regular expressions that can be used with the **case** command.

1. Log in as **user11** or as an ordinary user
2. Open the Text Editor.
3. Open a Terminal window.

4. Type the following script:

```
#!/bin/sh
echo "who makes your car? \c"
read mycar
case "$mycar" in
[Ff]o*)
 echo "An American car"
 echo "Made in Detroit"
 ;;
[Mm]erced*)
 echo "Yeah, sure you own one"
 ;;
[Hh]yun???)
 echo "An Asian car"
 echo "Made in Korea"
 ;;
*)
 echo "Never heard of that car"
esac
```

5. Save the script with the name `myscript22`
6. Type the command `chmod 777 myscript22`
7. Type the command `./myscript22`

The `case` statement uses several types of "wild cards" when doing comparisons:

- The asterisk (\*) matches any character or string of characters. It also matches "nothing"—if no characters are typed at the position where it occurs.
- The question mark (?) matches any single character.
- Enclosing more than one character in square brackets ([ ]) tells the `case` statement to accept any of the characters between the square brackets

By using these wild cards, this script accepts a wider variety of input:

- The **Ford** selection from the previous script was replaced with `[Ff]o*`. This lets the user type **f** or **F** as the first letter. The second letter has to be an **o**. The asterisk means that any character or characters will be accepted from the third character onward (or that the user can type nothing here, leaving the input at two characters).
- The **Mercedes** selection was replaced with `[Mm]erced*`. This lets the user type in **M** or **m** as the first character. The next five characters must be **erced** (lower case). After that, any character or string of characters will be accepted, or the user can type nothing here.
- The **Hyundai** selection was replaced with `[Hh]yun???)`. This lets the user type **H** or **h** as the first character. The next three characters must be **yun** (lower case). The three question marks (?) at the end of the script let the user type in any combination of **exactly** three characters.

The wild cards in this script are not intended to be "perfect." For example, notice that for the **Ford** selection, **folderol** would be accepted, and that for the **Mercedes** choice, **merced** would be accepted, but **MERCEDES** would not. Experiment with different combinations to make the case command accept and incorrectly typed answers in the way that you want it to.

## Creating a Selection Menu with the Case Command

The easiest way to make a menu in a Bourne shell script is with the **case** command. Try not to make a menu that is too long for terminal connections. Limit the size of the menu to about 20 lines down to make sure even old terminals can use the menu.

### Lesson 12.23 Creating a Menu in the Bourne Shell Script

In this lesson, an onscreen menu is created to guide the user.

1. Log in as **user11** or as an ordinary user
2. Open the Text Editor.
3. Open a Terminal window.
4. Type the following script:

```
#!/bin/sh
echo "select your vehicle "
cat << MENUEND
 1) Ford
 2) Mercedes
 3) Hyundai
MENUEND

read mycar

case "$mycar" in
 1)
 echo "An American car"
 echo "Made in Detroit"
 ;;
 2)
 echo "Yeah, sure you own one!"
 ;;
 3)
 echo "An Asian car"
 echo "Made in Korea"
 ;;
 *)
 echo "Never heard of that car"
esac
```

5. Save the script with the name **myscript23**
6. Type the command **chmod 777 myscript23**
7. Type the command **./myscript23**

*This script uses the text between **cat << MENUEND** and **MENUEND** to present a menu. By requiring the reader to enter a number rather than typing an answer, this script avoids problems with incorrect input. This script checks for the numbers **1**, **2** or **3**, rather than checking for the names **Ford**, **Mercedes** or **Hyundai**.*

### The for Command

The **for** command is good for processing lists of names or lists of anything. The **for** command follows the following syntax:

```
for variable in <word list>
do
```

```
statements
done
```

## Lesson 12.24 Using the **for** command

In this lesson the **for** command is used to read in a list of animals and to use that list to populate a series of echo commands.

1. Log in as **user11** or as an ordinary user
2. Open the Text Editor.
3. Open a Terminal window.
4. Type the following script:

```
#!/bin/sh
for animals in dogs cats mice
do
 echo "$animals have four legs"
done
```
5. Save the script with the name **myscript24**
6. Type the command **chmod 777 myscript24**
7. Type the command **./myscript24**

*The following output will appear:*

```
dogs have four legs
cats have four legs
mice have four legs
```

This example only echoes text to the screen. Of course, any type of operation can be put in the loop. For example, the **for** command is good for situations when a system administrator needs to perform the same repetitive task on some files and does not want to type in the same values again and again.

## Lesson 12.25 Using the **for** Command With a Repetitive Command

In this lesson, the **for** command is used to update a text files. One unique feature of the **for** command is it's unique ability to take words from a command run within the grave accent ( ` ) characters. Remember that the grave accent character ( ` ) runs the command. In the example, the **cat** command is run to change the list of names into a word list that the **for** command needs.

On a Sun keyboard, the grave accent character is the key just above the Back Space Key. On a PC keyboard, this key is usually above the Tab key on the left side of the keyboard, on the same key as the tilde (~) key.

1. Log in as **user11** or as an ordinary user
2. Open the Text Editor.
3. Open a Terminal window.
4. Create the following text file

```
Michael
Karen
Tom
Steven
```

5. Save the file as **namelists**
6. Create the following text file

**This is the system administrator.  
I have put this text file in your home  
directory to let you know that you  
are using too much disk space**

7. Save the file as **warnings**
8. Create the following shell script:

```
#!/bin/sh
for a in `cat namelists`
do
 cp warnings warningt
 echo "please delete some files $a" >> warningt
 cp warningt diskwarning.$a
done
rm warningt
```

9. Save the script as **myscript25**
10. Type the command **chmod 777 myscript25**
11. Type the command **./myscript25**

*Each time through the **for** loop, this script reads a name from **namelists**. It then creates a file called **diskwarning.<user's name>**. When it finishes, the files **diskwarning.Michael**, **diskwarning.Karen**, **diskwarning.Tom** and **diskwarning.Karen** have been created.*

*The files are saved in the same directory as the script. The script could have had the line **cp warningt /usr/\$a/diskwarning.\$a** to place the warning message in the user's home directory.*

*Here is the text of **diskwarning.Karen**:*

**This is the system administrator.  
I have put this text file in your home  
directory to let you know that you  
are using too much disk space**

**please delete some files Karen**

*(The file **warningt** is used as a temporary file so that each user's message file will not have the previous user's name in it.)*

12. To remove the warning messages, type the command **rm diskwarnin\***

*This example was created to illustrate how to use the **for** command for administration. In a real-world situation, the system administrator would make the warning messages appear on the users' screens or would send the warnings as messages.*

## Using the **while** command

The **while** Control-Flow command tests a condition. The condition is checked at the start. As long as the condition is **true**, the loop between the **do** and the **done** statements will continue to run. When the test condition becomes **false**, the loop terminates and the script continues with the next line. If the condition is **false** when the **while** statement is first encountered, the loop is not executed at all.

In a nutshell:

- The **while** tests a condition.
- If the condition is initially **true**, the loop runs until the test condition becomes **false**.
- If the test condition is initially **false**, the loop does not run at all.

The **while** command has the following syntax:

```
while (test condition)
do
 statements
done
```

## Lesson 12.26 Using the **while** Command

In this example, the **while** command is used to create a loop in which the script asks the user to type in a letter. The letter is then displayed. When the letter **z** is typed, the loop ends, and the script prints a message before terminating.

1. Log in as **user11** or an ordinary user.
2. Open the Text Editor.
3. Open a Terminal window.
4. Type the following script:

```
#!/bin/sh
echo "Type in a guess letter \c"
read myvar
while ["$myvar" != "z"]
do
 echo " you typed $myvar "
 read myvar
done
echo "That was the guess letter !"
```

5. Save the script with the name **myscript26**
6. Type the command **chmod 777 myscript26**
7. Type the command **./myscript26**

*This script is very easy to understand. Within the **while** loop, it simply asks the user to type in a letter (and press the Return key!). It checks to see if the letter is not z (!= "z"). If the user types anything other than z (lower case), the loop tells the reader what character was typed.*

*Notice that the line **echo "Type in a guess letter \c"** is not inside the loop. Because of this, the script only prompts the user to enter a character the first time. After that, the user must remember to type in a new character, without being prompted.*

*Now modify the script so it looks like this*

```
#!/bin/sh
while ["$myvar" != "z"]
do
 echo "type in a guess letter \c"
 read myvar
 if [-n "$myvar"]
then
 echo " you typed $myvar "
```

- ```

fi
done
echo "That was the guess letter !

```
8. Save the script with the name `myscript26.2`
 9. Type the command `chmod 777 myscript26.2`
 10. Type the command `./myscript26.2`

This script improves on the previous one in a couple of ways:

- Because the `echo "type in a guess letter \c"` statement is inside the loop, it prompts the user to enter a character each time through the loop.
- This script uses the `if` statement nested inside the `while` statement for some sanity checking. If the user presses the **Return** key without typing in a letter the statement `"you typed <letter>"` is not displayed. This happens because the statement `if [-n "$myvar"]` means "if the value of `$myvar` is not blank."

This shows that control statements can be nested within each other.

Using the while Command for Numerical Loops

The `while` command can be used when a list of numbers needs to be generated. Remember the while conditional test stops when the loop comes back up to the top and the condition is false.

Lesson 12.27 Using the while Command to Creating a List of Numbers

In this lesson the while command creates a list of numbers from 5 to 15.

1. Login as user11 or an ordinary user
2. Open the Text Editor.
3. Open a Terminal window.
4. Type the following script:

```

#!/bin/sh
numvar=4
while [ $numvar != 15 ]
do
numvar=`expr $numvar + 1`
echo " The number is $numvar"
done

```
5. Save the script with the name `myscript27`
6. Type the command `chmod 777 myscript27`
7. Type the command `./myscript27`

The results are pretty obvious. This script prints the values of `numvar` from 5 to 15:

```

The number is 5
The number is 6

```

```

The number is 15

```

When the `while` loop starts, the value of `numvar` is 4. Each time through, the loop increases the value of `numvar` by 1 and prints its value. When `numvar` reaches 15, the test condition `$numvar != 15` is false, and the loop stops.

The `until` Command

The `until` test condition is the exact opposite of the `while` test condition. The `until` Control-Flow command tests at the start of the loop to see if its condition is **false**. If the condition is **false**, the script executes the loop between the `do` and `done` statements until its test condition becomes **true**. At this point, the script continues with the next statement after the `done` statement. If the condition is **true** when the `until` statement is first encountered, the loop will not be executed.

In a nutshell:

- The `until` command tests a condition.
- If the test condition is initially **false**, the loop runs until the test condition becomes **true**.
- If the test condition is initially **true**, the loop does not run at all.

The `until` Control-Flow command supports the following format:

```
until (test condition)
do
    statements
done
```

Lesson 12.28 Using the `until` Command to Generate Numbers

The `until` command is used in this example for a simple loop demonstration. The number 5 is given to a variable. The script runs until the variable reaches 16.

1. Log in as `user11` or as an ordinary user.
2. Open the Text Editor.
3. Open a Terminal window.
4. Type the following script:

```
#!/bin/sh
numvar=5
until [ $numvar != 16 ]
do
    echo " The number is $numvar"
    numvar=`expr $numvar + 1`
done
```

5. Save the script with the name `myscript28`
6. Type the command `chmod 777 myscript28`
7. Type the command `./myscript28`

*This script produces no output! Why? Because the test condition is initially **true**. When the `until` statement is executed for the first time, the value of `numvar` is 5. The statement `$numvar != 16` is **true** here, so the loop will not run!*

8. Type in the following script

```
#!/bin/sh
numvar=5
until [ $numvar -gt 15 ]
do
    echo " The number is $numvar"
    numvar=`expr $numvar + 1`
done
```

9. Save the script with the name `myscript28.2`
10. Type the command `chmod 777 myscript28.2`
11. Type the command `./myscript28.2`

In this script, the value of **numvar** is 5 the first time through the loop. The script prints **The number is 5** the first time. It then continues to print the number until it prints **The number is 15**.

When the **numvar** variable reaches 16 at the bottom of the loop, the **until** command goes to the top of the loop and checks the condition again. Now, **\$numvar -gt 15** is **true**, and the loop stops. The **until** command always evaluates the test condition at the top of the loop.

The break command

The **break** command is used to break out of a **for**, **while** or **until** loop, regardless of the test condition. The script then continues to run after the **done** keyword. The **exit** command is used to leave a script immediately. Don't confuse the **exit** command with the **break** command.

Lesson 12.29 Using the break Command

In this lesson the loop will run continuously until the **break** command is reached. The **break** command is inside of an **if** statement that checks the variable **goodbye** for the answer **y**. If the user types **y** the **break** command is executed, and the loop is terminated immediately, without testing its test condition.

1. Log in as **user11** or as an ordinary user
2. Open the Text Editor.
3. Open a Terminal window.
4. Type the following script:

```
#!/bin/sh
numvar=5
while [ $numvar != 16 ]
do
    echo " Type y and press Return to leave the loop, any other key to stay in "
    read mykeystroke
    if [ "$mykeystroke" = "y" ]
    then
        echo "break out of the loop"
        break
    fi
    echo "loop running again"
done
```

5. Save the script with the name **myscript29**
6. Type the command **chmod 777 myscript29**
7. Type the command **./myscript29**
This script breaks out of the loop when the lower case y key is pressed.

The continue command

The **continue** command returns to the top of the loop, ignoring anything that is below the **continue** command. This can be used on **for**, **while** and **until** loops.

Lesson 12.30 Using the continue Command

In this lesson the loop will run continuously until the **break** command is reached. The **break** command is inside an **if** statement that checks the variable **goodbye** for the answer "**y**".

1. Log in as **user11** or an ordinary user.
2. Open the Text Editor.
3. Open a Terminal window.
4. Type the following script:

```
#!/bin/sh
numvar=5
while [ $numvar != 16 ]
do
    echo "top of the loop"
    echo " Type any letter except - y to leave - c continue command "
    echo "press Return after letter"
    read mykeystroke
    if [ "$mykeystroke" = "y" ]
    then
        echo "break out of the loop"
        break
    fi
    if [ "$mykeystroke" = "c" ]
    then
        echo "continue to top of loop"
        continue
    fi
    echo "bottom of the loop "
done
```

5. Save the script with the name **myscript30**
6. Type the command **chmod 777 myscript30**
7. Type the command **./myscript30**
*When the **c** key is pressed, the script immediately returns to the top of the loop, without finishing the rest of the loop, because of the **continue** statement Also, this script breaks out of the **while** loop when the **y** key is pressed. The **bottom of the loop** statement does not appear at that point.*

Debugging Bourne Shell Scripts

Sometimes Bourne shell scripts can become very complex. When a Bourne shell script becomes complex, there are some limited debugging utilities that can be used.

set -x A line used in the script to start the debugging

set +x A line used in the script to end the debugging

Lesson 12.31 Using **sh -v** and **set -x** and **set +x**

In this lesson the **set -x** and **set +x** commands are used to check the **myscript30** script as it is run.

1. Log in as **user11** or as an ordinary user.
2. Open the Text Editor.
3. Open a Terminal window.
4. Type the command **cp myscript30 myscript31**
This command copies myscript30 to myscript31.
5. Type the command **./myscript31**

The **myscript31** script will run. Notice that each line of the script is displayed as it is being run.

6. Modify the **myscript31** script so it looks like this:

```
#!/bin/sh
set -x
numvar=5
while [ $numvar != 16 ]
do
    echo " The number is $numvar"
    numvar=`expr $numvar + 1`
done
set +x
myvar327=83
```

7. Type the command **./myscript31**

When the script runs, the results of each line between the **set -x** and **set +x** statements are shown as that line is executed. Here is what appears near the end of the loop:

```
numvar=15
+ [ 15 != 16 ]
+ echo The number is 15
  The number is 15
+ expr 15 + 1
numvar=16
+ [ 16 != 16 ]
+ set +x
```

Any time a script has problems just add the line **set -x** just above the suspected problem area. The script will be displayed line for line as it is being processed. The traced lines are shown with a plus (+) sign.

Notice that the statement **myvar327=83** is not displayed as it is executed. This is because it is after the **set +x** statement. This line is in the script just to show this. Setting the variable **myvar327** has no real purpose..

Lesson 12.32 Create a Custom Utility

Most system administrators have their personal “bag of tricks” when it comes to custom-made commands. This lesson shows how to create a custom-made command to automatically list directories with the **ls -l** command. The command should be put in the **/usr/bin** directory, because all users have this in the **PATH** variable. Any command in the **PATH** variable’s list will automatically be executed when typed at a terminal or console.

1. Log in as the root user.
2. Open the Text Editor.
3. Open a Terminal window.
4. Type the following script:

```
#!/bin/sh
ls -l
```

Notice that the last line is **ls-l** (small **L**).

5. Save the script with the name **longlist**
The **longlist** command displays a long listing of files and directories with the **ls -l** (small **L**) command.
6. Type the command **chmod 555 longlist**
7. The **chmod 555** sets read/execute permissions on the shell script **longlist**
8. Type the command **cp longlist /usr/bin/**

The `/usr/bin` directory is a directory that all the users have in their `PATH` variable. The command `longlist` will be available for all the users when they type the command `longlist` in a terminal.

9. Log off as root.
10. Log in as user11.
11. Type the command `longlist`

The command `longlist` now acts the same as the `ls -l` command. It is available because the shell script `longlist` is in the `/usr/bin` directory. If all users need to use a command, put the command in the `/usr/bin` directory.

Bourne Run Control Scripts

Bourne shell scripts are used all throughout the Solaris 9 operating system. When Solaris 9 starts, it executes what are known as Run Control Scripts. These scripts perform all of the housekeeping functions after the `init` process is started. For example, these scripts start the Apache web server, the CDE desktop, and start the `sendmail` daemon.

These scripts are located in the following directories:
`/etc/rc0.d, /etc/rc1.d, /etc/rc2.d /etc/rc3.d rcs.d`

Run Control Script names start with an `S` or a `K`. The scripts that start with an `S` are used to start something. For example, in the `/etc/rc3.d` directory are the scripts `S90samba` and `S50apache`. The `S90samba` script is used to start the Samba daemon `smbd` to allow the Samba service to run. The `S50apache` script is used to start the Apache web server. The script `/etc/rc1.d/K16apache` is used to kill the Apache web server.

Lesson 12.33 Create a Run Control Script

In this lesson, readers create a relatively harmless script that displays the message “my shell script from Chapter 12” when the system starts. The shell script is tested before the server reboots, so that it will not hang during the boot process. This could prevent the server from working.

1. Log in as the root user.
2. Open the Text Editor.
3. Open a Terminal window.
4. Create the script
`echo “my shell script from Chapter 12”`
`sleep 5`
5. Save the file as `/etc/rc3.d/S25myscript`
6. Type the command `chmod 777 /etc/rc3.d/S25myscript`
7. Type the command `/etc/rc3.d/S25myscript`

MAKE SURE THE SCRIPT IS CORRECT !!! IT SHOULD ONLY ECHO THE FIRST LINE OF THE SCRIPT ON THE TERMINAL, AND THEN SLEEP FOR 5 SECONDS BEFORE THE SYSTEM PROMPT REAPPEARS.

SOLARIS 9 COULD HAVE PROBLEMS STARTING IF YOU MAKE A MISTAKE !!!

Before a server is rebooted or shut down, make sure that any new run control scripts work. If the run control script hangs, the server will not start properly.

8. Type the command `init 6`
Watch the startup messages carefully. The message `my shell script from Chapter 12` should appear.
9. Log in as a root user.
10. Open a Terminal window.

11. Type the command `rm /etc/rc3.d/S25myscript`
The command `rm /etc/rc3.d/S25myscript` removes the `S25myscript` from the system. If you forget to remove this script, the `my shell script from Chapter 12` message will appear each time you boot the system.

Users `.profile` File

Users of the Bourne shell use the `.profile` file to set their own environmental variables. This script can not be seen with the `ls` command because it is a hidden file. Any file that starts with a period (`.`) is a hidden file that will not be seen by the `ls` command. However, the `ls -A` command will show all the files in a directory, even hidden files.

Lesson 12.34 Working With the `.profile` File

The `.profile` file is used to set a user's environmental variables. In this lesson readers will set the variable `DOG` to the name of the author's dog, `zeppiman`. This variable will be available to the user after the user logs off the system and then logs back on.

1. Log in as `user11`.
2. Open a Terminal window.
3. Open the Text Editor.
4. Edit the file `/usr/user11/.profile`
Add the following lines to any lines that already exist in the file:
`DOG=zeppiman`
`export DOG`
5. Save the file as `.profile`
6. Type the command
`./profile`
This command is "dot-space-dot-front slash-dot" followed by "profile"
7. Type the command `echo $DOG`
The variable `DOG` should be set to `zeppiman`.
8. Log off the system
9. Log back into the system as `user11`.
10. Open a Terminal window.
11. Type the command `echo $DOG`
The variable `DOG` should still be set to `zeppiman`

Lesson 12.35 Capturing a Key

In the previous lessons the user had to press the Return key to enter in a value. To make a more impressive shell script, create the following script, which lets users enter values without pressing Return.

Please note that there might be some problems if a user connects through a terminal session and the user's terminal is not fully recognized by Solaris 9. This script was tested from the keyboard of a SunBlade100 and in a telnet session from Microsoft Windows.

1. Log in as `user11` or an ordinary user.
2. Open the Text Editor.
3. Open a Terminal window.
4. Type the following script:

```
#!/bin/sh
CatchKey () {
```

```

trap "" 2 3
oldSttySettings=`stty -g` # use grave accent marks not single quotes
stty -echo raw
echo "`dd count=1 2> /dev/null`" # grave accent marks here, not single quotes
stty $oldSttySettings
trap 2 3
}
PressKey=""
while [ "$PressKey" != "x" ]
do
echo "press the x key to exit \c"
PressKey=`CatchKey` # these are grave accent marks not single quotes
echo "You just typed $PressKey"
#
# set a variable to anything needed
#
myvar1=$PressKey
echo "myvar1 is set to $myvar1"
# Add your script material here
# Include the case command to make a menu
done

```

5. Save the script with the name `myscript35`
6. Type the command `chmod 777 myscript35`
7. Type the command `./myscript35`

The `CatchKey()` function uses the `trap` command and the `stty` command to capture the keystrokes from the keyboard. The `PressKey` variable retrieves the value of the function `CatchKey()`. Because the keys need to be re-read, the `CatchKey()` function needs to be within a `while` loop. The value inside the `$PressKey` variable can be used for any typical duties of a variable. For example, copy the contents of the `PressKey` variable into other variables. Use the secondary variables for nested commands, such as `if`, `while` and `until`.

Key Points to Remember

The Bourne shell is the default system administration shell available on Solaris 9. All of the run control scripts and most software scripts are written in the Bourne shell. It will take time and patience to learn how to make nice Bourne Shell scripts. Follow the examples in this chapter. Later, when you become more comfortable with Bourne shells, modify the scripts you created from this chapter to fit your needs. Make tiny changes to the scripts and test the scripts often. Do not make a huge, complex script and then try to debug the script. That is the worst way to do things.

Chapter 13 Working with Hard Drives

Lessons in This Chapter

Lesson 13.1 Using the <code>format</code> Utility to View a disk's Partitions.....	13-8
Lesson 13.2 How To Add a New Hard Drive.....	13-7
Lesson 13.3 Format the New Hard Drive.....	13-12

Introduction

To follow the lessons in this chapter, your system needs to have a second hard drive. Understand that the **format** command will work on a system with only one hard drive, but the data on that hard drive will be destroyed after the reformatting operation. It is very important that the reader understand that when a drive is reformatted (even modifying unused slices), all of its information on the hard drive is destroyed. To better understand this concept, ask yourself the simple question: How can a hard drive that is running the operating system be reformatted? The partitions are rebuilt and the original files are destroyed. For the lessons in this chapter, any second hard drive, even a little 100 MB drive, will work. Most used computer stores sell this type of small drive for less than \$10 apiece.

This chapter shows how to reformat a hard drive and create a new file system. It also covers how to *mount* the reformatted hard drive after the new file system has been created.

Hard Drive Slices

A hard drive within a SUN Solaris system can have up to eight individual partitions. Another name for a partition is a *slice*. Each partition holds a separate file system. These file systems can be regular UFS file systems or special file systems, like the **swap** or **procfs** file systems. The **swap** partition is optimized for swapping memory pages out of physical memory and onto a disk. The **procfs** file system is used to store images of processes. If one file system becomes corrupted it will not affect another file system.

Sun Solaris uses the following convention with slices:

- 0 / The first slice is the zero 0 slice. This slice always holds the root (/) file system.
- 1 swap partition This slice is optional. All hard drive swapping occurs on this slice.
- 2 entire disk This is a symbolic slice that represents the entire disk. This slice does not use any physical space on the hard drive. It is used primarily by backup and restore programs to copy entire hard drive contents.
- 3 myslice3 This slice is available for customer use.
- 4 myslice4 This slice is available for customer use.
- 5 /opt Most third-party software installs on this slice.

6 /usr Native Solaris programs and commands are ordinarily stored here.

7 /export/home The user's home directories are stored on this slice.

The Swap Partition Slice 1 and Disk Image Slice 2

Slice 1 swap partition and Slice 2 [entire disk] are special slices that do not function like the other partitions on the hard drive. These slices can not be used for ordinary day-to-day tasks. It is not possible to directly mount and copy files onto slice 1 or slice 2.

Slice 1 is used by the operating system to swap chunks of memory out of the physical RAM chips and onto the hard drive. This usually occurs if the demand for memory by a software program exceeds the amount of physical memory installed on a system. In essence, the operating system treats slice 1 like a big, slow memory chip. When a system copies memory to the swap partition, the system usually becomes very slow and sluggish. This is because read/write operations to a hard drive are very slow when compared to memory read/writes.

Slice 2 represents the entire hard drive. This is only a symbolic slice that does not take any space off of the hard drive. Slice 2 is a slice that has special rules and restrictions. Ordinary users can not mount this slice, and they can not copy data from or to it. A system administrator can mount this slice but not use it for copying data. Under some special circumstances, the root user can mount this slice to copy and restore files from a tape drive or other backup device. As can be seen below in the modified screen shot of a hard drive (taken from the Solaris Management Console), slice 2 represents the entire disk, from Cylinder 0 to cylinder 10669.

Disk Layout:



Backup and restore programs use slice 2 to copy the entire contents of the hard drive to and from backup devices. This slice can only be accessed in block mode or what is known as raw mode. If the slice is mounted in raw mode, no files can be seen. All files and directories are linked and represented by this slice.

Sun Solaris has special rules on how slices and disks are used. For the boot disk, but not for other disks, the entire disk is represented by slice 2, and this slice is not available for any other purpose.

The following rules apply only to the boot disk:

1. The root file system must reside on slice 0. This slice is not available for any other purpose.
2. An administrator can create a system with any combination of slices and directories, except for slice 0 and slice 2. If for some reason an administrator wants to put the swap partition on slice 7, that is possible. If an administrator want to put the `/opt` directory under slice 1, that is also possible (provided that administrator disables the swap partition or moves it to another location). Basically, anything goes, except with slice 0 and slice 2.
3. A system does not need the swap partition (slice 1) because a swap file could be used instead of a swap slice. Slice 1 can be used for something other than swapping files between the hard drive and memory. However, this is dangerous because if the system runs out of physical memory, it will crash. Most system administrators follow the convention of using slice 1 as a swap partition, even though this is not required.

- The `/opt`, `/usr`, and `/export/home` directories do not need to have a separate slice dedicated to each of them. If an administrator installs a SUN Solaris system and does not allocate a slice for `/opt`, `/usr`, and `/export/home` directories, the install program will create ordinary directories in the root file system named `/opt`, `/usr` and `/export/home`. It will then use these directories for their intended purposes. However, for performance reasons, most system administrators do create a separate slice for each of these directories. When a directory is mounted on a separate slice, the read/write operations occur faster under this directory.

What Makes Up a Hard Drive and How That Relates to Solaris 9

A hard drive has four major components. On the outside of a hard drive are the two connectors, one for the power supply and one for the IDE or SCSI cable. Figure 13.1 shows the back end of an IDE hard drive. The cable connector is on the left side of the photo, and the power supply connector is on the right. In the middle are jumper pins (with a blue jumper on two pins). The placement of the jumpers defines the hard drive as a slave or a master drive.



Figure 13.1 IDE Hard Drive Connectors

The power connector and IDE or SCSI connections are pretty much “goof proof.” The cables are slotted and can only be inserted one way. The only component that can be modified is the jumpers. As mentioned above, a

jumper is a small plastic connector  that straddles two pins and is used to change the hard drive’s settings. Figure 13.2 shows the jumper label found on the author's hard drive.

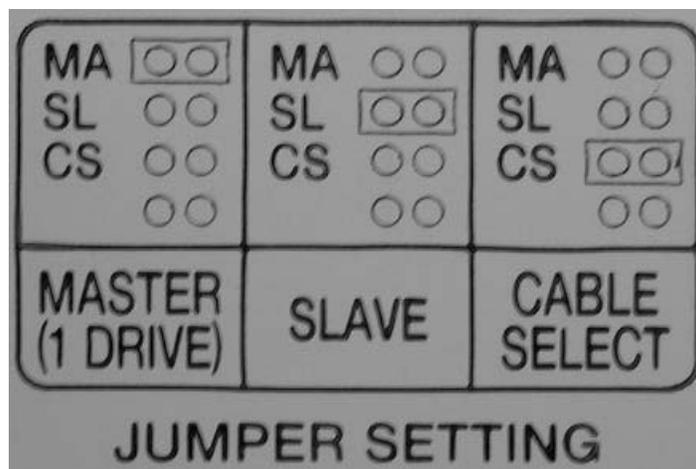


Figure 13.2 Jumper on an IDE Hard Drive

A similar label should appear on one of the stickers on the top of your hard drive. To install a second hard drive, change the jumpers on the second hard drive to the SLAVE setting or SL. Also, make sure to change the jumpers on the original hard drive to MA or MASTER.

After the jumpers are set, power on the system. When the **OK** prompt is displayed, type the command **probe-ide** or **probe-scsi**, depending on your setup. The OpenBoot firmware should now see the second hard drive.

If it does not, the operating system definitely will *not* be able to work with the hard drive. The basic rule is "If hardware is not detected at the firmware level, it will *not* be detected inside the operating system." The only exception to this rule is if a special device driver is needed before the OS can *see* a device. However, it is extremely rare that a hard drive or CDROM needs a device driver to operate.

On the bottom of the hard drive is the on-board disk controller. This is shown in Figure 13.3.



Figure 13.3 On-board Hard Disk Controller

This is not the built-in IDE or SCSI controller that comes with a Sun system. This controller is on the hard drive and was made by the manufacturer to work with the various components inside the hard drive.

On the inside of the hard drive is a component called the *platter*. Like the mylar surface of a floppy diskette, the ferromagnetic surface of the platter can be magnetized by the drive's magnetic read/write head to hold the bits and bytes that make up the system's data.

The magnetic head is attached to an actuator arm that arcs toward and away from the center of the disk. Figure 13.4 is an illustration of an actuator arm, disk head and platter.

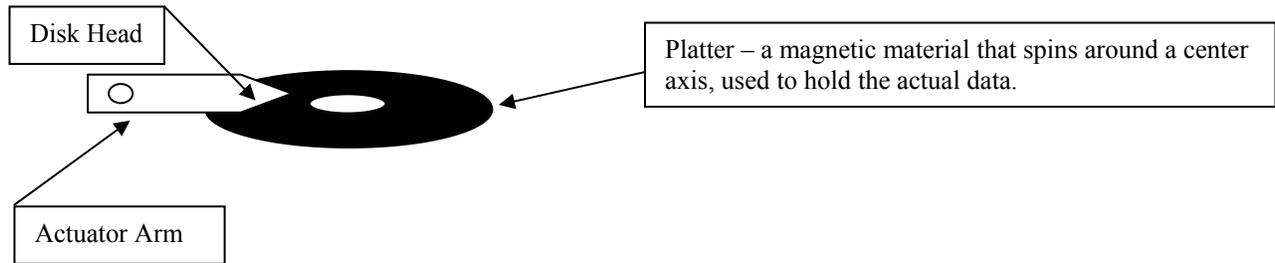


Figure 13.4 Actuator Arm, Disk Head and Platter

Figure 13.5 shows the same components on an actual IDE hard drive.

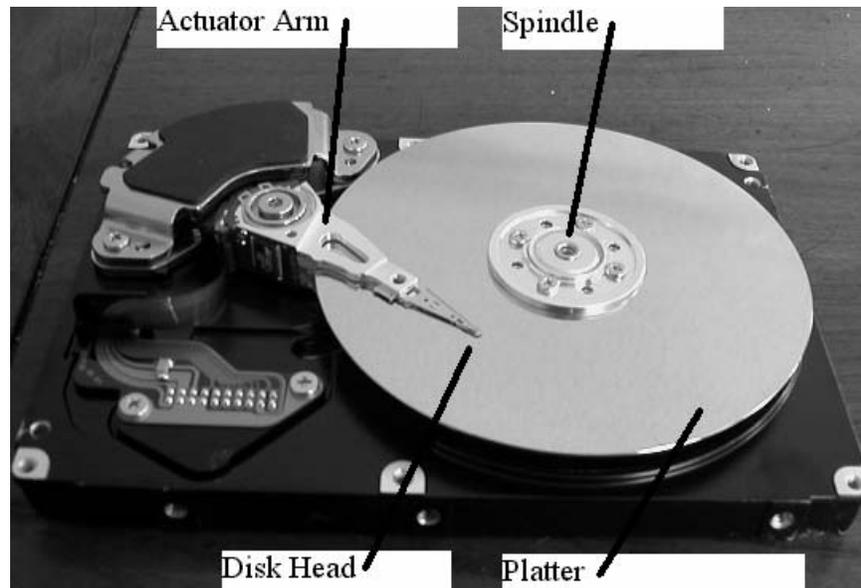


Figure 13.5 IDE Hard Drive Internal Components

Although a (broken) hard drive was disassembled for the figure shown above, never remove internal pieces from a hard drive. Understand that hard drives are manufactured in clean rooms. Once the platter and disk heads are exposed to *dirty air*, the hard drive becomes useless. The magnetic heads travel only a couple of microns above the disk surface. Dust, and even a human hair, will destroy the hard drive's disk surface.

If a hard drive stops working, call the manufacturer to see if it is still under warranty. The only possible repair that can be made on a hard drive is to find a second broken hard drive and swap the on-board disk controller. If the surface of the hard drive or the heads go bad, throw the hard drive away. It can not be fixed by most individuals. Just make sure to save the on-board disk controller so that it can be used on the exact same model of hard drive in the future.

Hard drives usually have several disks that are stacked above each other. These disks are attached to a center post, or *spindle*. Figure 13.6 shows multiple platters and a spindle.

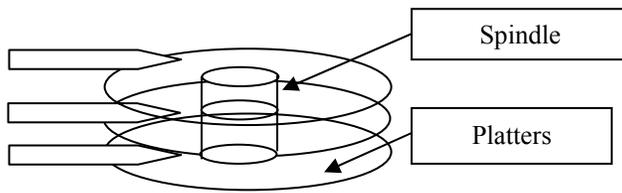


Figure 13.6 Multiple Platters and a Spindle

The entire assembly of platters and spindle is spun around by an electric engine that is embedded inside the spindle.

A *track* is the invisible circular path that a disk head makes around a platter as the platter spins. A *cylinder* is a collection of all the tracks that are in the same position on different platters.

Figure 13.7 is an illustration of a track and cylinder. The track can be broken down into smaller sections called *sectors*. A sector is just like a pizza slice; it divides a track into smaller pieces.

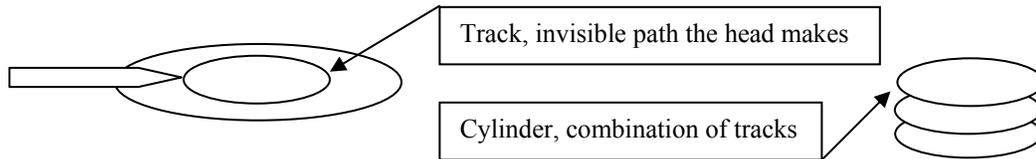


Figure 13.7 Track and Cylinder

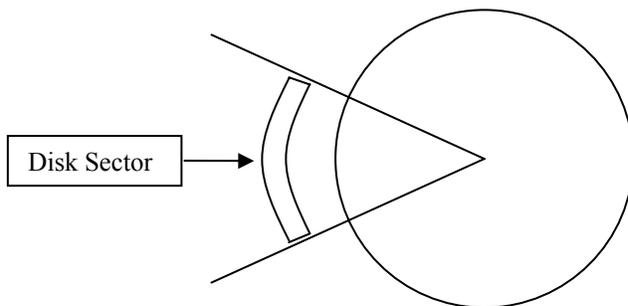


Figure 13.8 Illustration of a sector

In Solaris 9, a *slice* is simply a collection of cylinders, just as a cylinder is a collection of tracks. Sectors in Solaris 9 are 512-byte blocks of data. Slice 0, the root (/) slice, is always on the outside of the disk, where data can be read the fastest. Slice 7 is always the slice closest to the center of the disk.

There are two types of disk controllers that are widely used with computers, SCSI (Small Computer Systems Interface) controllers and IDE controllers. IDE controllers are typically found on Intel-based computers and workstations. SCSI controllers and SCSI disks are usually found on high end servers.

The standard IDE controller can manage four disks. The first two disks are under the primary IDE interface, and the second two disks are under the secondary IDE interface. Under each interface there is a master disk and a slave disk. The four possible disk types are:

Primary IDE interface master disk
Primary IDE interface slave disk
Secondary IDE interface master disk
Secondary IDE interface slave disk

Some high end servers have more than one IDE controller and can have eight disks:

First IDE disk controller:

Primary IDE interface master disk
Primary IDE interface slave disk
Secondary IDE interface master disk
Secondary IDE interface master disk

Second IDE disk controller:

Primary IDE interface master disk
Primary IDE interface slave disk
Secondary IDE interface master disk
Secondary IDE interface slave disk

Just understand that an IDE controller can handle a maximum of four disks. However, more than one IDE controller can be added to a computer.

Lesson 13.1 How to Add a New Hard Drive

Some Sun equipment can be installed without the need to reboot the operating system. Other pieces of equipment can be installed while the server and operating system stay up. It depends on what server and what equipment is being attached to each other.

When a new hard drive is installed, the system must be rebooted. To reboot the system, perform these steps:

1. Log in as the root user.
2. Open a Terminal window.
To open a terminal window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Tools menu item, then left click on the Terminal icon.
3. Type the command **touch /reconfigure**
This tells the Solaris operating system and SPARC hardware to look for new hardware.
4. Type the command **init 5**
*This will power down the system after the operating system gracefully quits. Some older systems do not support the **init 5** option. On these systems, type the command **init 0** and then type the command **power-off** at the **OK** prompt)*
5. Make sure the system is powered down.
6. Install the new hard drive(s).
7. Power on the system.
8. If the command **touch /reconfigure** was not typed at the command line (step 3), when the server starts up again, immediately press the STOP + A key combination. Then type the command **boot -r** at the **OK** prompt
9. Log in as the root user.
10. Open a Terminal window.
11. Type the command **prtconf** to see if the new hardware has been recognized by the operating system. *If the **prtconf** command recognizes the hard drive, the drive should be recognized by the **format** command.*

After a new hard drive is added, the system administrator can use the **format** utility to view its partitions. This is covered in the following lesson.

Lesson 13.2 Using the **format** Utility to View a Disk's Partitions

The **format** command is used to change the slices on a disk. Unfortunately, Microsoft also has a **format** command in MS-DOS and Windows that does something completely different. The **fdisk** utility in MS-DOS and Windows is the closest thing to the **format** command in Solaris 9. In this example the **format** command is only being used to show the slices on a computer hard drive.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command **format**

When the format command is typed, the screen displays the output

```
Searching for disks...done
AVAILABLE DISK SELECTIONS:
0.      c0t0d0 <ST315320A cyl 29649 alt 2 hd 16 sec 63>
        /pci@1f,0/ide@d/dad@0,0
1.      c0t2d0 <ST315320A cyl 29649 alt 2 hd 16 sec 63>
        /pci@1f,0/ide@d/dad@2,0
Specify disk (enter its number):
```

Select the second disk (if the workstation has a second disk, it will be numbered as 1.). If a second disk is not present, select the boot disk, usually c0t0d0s. Type the number that corresponds to the disk.

4. Type the command **partition**
5. Type the command **print**
6. The output of this command should look similar to this:

```
partition> print
Current partition table (original):
Total disk cylinders available: 10670 + 2 (reserved cylinders)
```

Part	Tag	Flag	Cylinders	Size	Blocks
0	root	wm	2218 - 10669	3.81GB	(8452/0/0) 7987140
1	swap	wu	0 - 1733	800.11MB	(1734/0/0) 1638630
2	backup	wm	0 - 10669	4.81GB	(10670/0/0) 10083150
3	unassigned	wm	2110 - 2217	49.83MB	(108/0/0) 102060
4	unassigned	wm	2002 - 2109	49.83MB	(108/0/0) 102060
5	unassigned	wm	1894 - 2001	49.83MB	(108/0/0) 102060
6	unassigned	wm	1786 - 1893	49.83MB	(108/0/0) 102060
7	unassigned	wm	1736 - 1785	23.07MB	(50/0/0) 47250

This is the partition table on the disk. The Cylinders column shows the beginning and ending cylinders of a slice. Partition 0 is the root partition. Notice that Partition 2 starts at cylinder 0 and ends at the last cylinder. Remember that Slice 2 represents the entire disk.

7. Type the command **quit**
This exits from the partition sub-menus.
8. Type the command **quit**
*This exits from the **format** command.*

Now that the **format** command has been used to specify the disk slices, how are the disks accessed? The hard drives and slices are accessed through files in the **/dev** directory with the **mount** command. Before describing

the **mount** command in detail it is critical that a reader understand the disk naming convention used by Sun Solaris. The **mount** command will be described later.

Solaris Disk Naming Conventions

For every hard drive known to the operating system, there is a file in the `/dev/dsk` directory that is used to access that drive. The convention used to name these files is `/dev/dsk/c#t#d0s#` (where `c#t#d0s#` is different for each hard drive).

/dev	This is the <code>/dev</code> or device directory.
/dev/dsk	This subdirectory contains files that represent all hard drives and slices on the system.
c#	This represents the first controller. /c0 – the first IDE controller /c1 – represents the second IDE controller if it exists /c2 – represents the third IDE controller if it exists
t#	The target drive. With an IDE system the disks are identified by: t0 – primary IDE controller, master drive t1 – primary IDE controller, slave drive t2 – secondary IDE controller, master drive t3 – secondary IDE controller, slave drive
d#	The device number, used with disk arrays (does not apply to standard IDE controllers). This number is always zero (0).
s#	The slice number – the slice to be accessed s0 – slice0 s1 – slice1 s2 – slice2 s3 – slice3 s4 – slice4 s5 – slice5 s6 – slice6 s7 – slice7

To access the root slice 0 on the master hard drive attached to the first IDE interface the system, an administrator would use the following file:

```
/dev/dsk/c0t0d0s0
```

To access the fifth slice on the slave drive attached to the second IDE controller, , an administrator would use the following file:

```
/dev/dsk/c1t4d0s5
```

When a server has a SCSI controller installed, the SCSI disks are accessed using the same `/dev` files. A SCSI controller can handle seven devices. These can be any combination of SCSI hard drives, SCSI CDROM drives, or any other SCSI device. As with IDE controllers, the convention used is: `/dev/dsk/c#t#d0s#` (where `c#t#d0s#` is different for each hard drive).

/dev	This is the /dev or device directory.
/dev/dsk	This subdirectory contains files that represent all hard drives on the system.
c#	This represents the first controller, /c0 – the first SCSI controller /c1 – the second SCSI controller if it exists. /c2 – the third SCSI controller if it exists.
t#	Target drive. With a SCSI system, this references the SCSI id. t0 – first SCSI device t1 – second SCSI device t2 – third SCSI device t3 – fourth SCSI device t4 – fifth SCSI device t5 – sixth SCSI device t6 – seventh SCSI device t7 – the SCSI controller (it also has a SCSI id)
d#	Device number, also known as the LUN (Logical Unit Number), used with disk storage devices. The manufacturer will specify this. It should be zero (0) by default.
s#	The slice number – the slice to be accessed. s0 – slice0 s1 – slice1 s2 – slice2 s3 – slice3 s4 – slice4 s5 – slice5 s6 – slice6 s7 – slice7

To access the fifth slice on the second disk on the third controller, an administrator would use the following file:

```
/dev/dsk/c2t1d0s5
```

To access the third slice on the first controller on the fifth disk, an administrator would use the following file:

```
/dev/dsk/c0t4d0s3
```

Logical and Physical Device Names

The **/dev/dsk/c#t#d#s#** naming convention is known as the *logical device name*. The **/devices** directory also has a very complex naming convention known as the *physical device name*. A physical device name looks something like:

```
/devices/pci@2c,0/pci@3,2/scsi@3/dad@0,0:a
```

Every file in the **/dev** directory (logical device names) has a corresponding file in the **/devices** directory (physical device names). The Kernel likes to work with the **/devices** names. But human beings find it easier to work with logical device names such as **/dev/dsk/c0t0d0s0**. Imagine if a system administrator would have to type a physical device name such as

```
/devices/pci@2c,0/pci@3,2/scsi@3/dad@0,0:a
```

every time he or she wanted to connect to a disk!

Basically the `/dev` directory has human friendly files and the `/devices` directory has files the Kernel likes to use. The `/dev` directory files are linked to `/devices` files.

The Kernel also likes to use a shorthand name for devices. This is known as the *instance name*. The instance name is stored in the `/etc/path_to_inst` text file.

This file should not be modified by the system administrator. There are utilities within the operating system that detect and modify this file. These utilities should be used to update the system when new hardware is installed on a server. Just know that the `/etc/path_to_inst` file saves the physical device names and their corresponding instance names.

As a review, every device has three names that correspond to it:

- Logical device names – saved under the `/dev` directory, look something like: `/dev/dsk/c0t2d0s4`
- Physical device names – saved under the `/devices` directory, look something like:
`/devices/pci@2c,0/pci@3,2/scsi@3/dad@0,0:a`
- Instance names – saved in the text file `/etc/path_to_inst`

The mount Command

Now that we've dealt with the various types of device names, it's time to explore the `mount` command. In Microsoft Windows, a hard drive or partition is accessed by a drive letter. For example, if a second hard drive is added to a PC, it might be represented by the `d:` drive letter. However, in UNIX there is no such thing as a drive letter. UNIX takes a different approach to accessing a hard drive or a slice on a hard drive. In UNIX, the `mount` command is used to associate a directory with a hard drive and slice.

Let's take the case where a second hard drive has been added to a workstation. In this example the workstation is powered on and the hard drive has been recognized by the operating system.

To access this disk, the system administrator creates a directory. It does not really matter what utility creates the directory, or what the director is called. In this case, the directory will be called `/mymountdirectory` and we will use `mkdir` command to create this directory.

```
mkdir /mymountdirectory
```

So far, `/mymountdirectory` is just an ordinary directory, nothing special!

Now the system administrator uses the `mount` command to access the second hard drive under the directory just created.

```
mount /dev/dsk/c0t2d0s0 /mymountdirectory
```

Now the directory `/mymountdirectory` takes on special significance. Any time a file is copied into the `/mymountdirectory` it is actually being copied onto the second hard drive.

There are various commands that can be used to troubleshoot a system if a hard drive or CDROM is not being recognized by the system. The **prtconf** command prints out the system information. This is a very useful tool if for some reason a hard drive or CDROM can not be used. It would indicate if the CDROM or hard drive was recognized on the hardware level by the system.

Some devices are what is know as “hot swappable,” meaning that they can be added to and removed from a server without the need to shut down the server or operating system. The command **drvconfig** is used to configure hot swappable devices.

There is a more recent version of the **drvconfig** command, called **devfsadm**. This utility is easier to understand and is more robust.

When a hard drive comes fresh from the factory it is unlikely that the hard drive will have the slices that a user wants. The drive must be prepared with the **format** utility. This program is covered in the next lesson.

Quick Tip

- Unfortunately, Microsoft has a very *different* utility that is also called **format**. This causes a lot of confusion!
- The Sun Solaris version of **format** does different things than the same command in MS-DOS or Windows. When trying to understand Solaris 9, put Microsoft’s **format** utility completely out of your head! This is very different in concept than the Sun Solaris **format** command!

Lesson 13.3 Format the New Hard Drive

In this lesson the reader will use the **format** command to format the hard drive.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command **format**

A message like the following should appear on the screen:

```
# format
Searching for disks...done
```

AVAILABLE DISK SELECTIONS:

- 0. c0t0d0 <WDC AC25100L cyl 10670 alt 2 hd 15 sec 63>
/pci@1f,0/ide@d/dad@0,0
- 1. c0t2d0 <WDC AC35100L cyl 10670 alt 2 hd 15 sec 63> dog
/pci@1f,0/ide@d/dad@2,0

Specify disk (enter its number):

This text can have a different look depending on the system’s hardware.

4. Type **1** (one) for the *second* disk. (In most cases it should be **1. c0t2d0**.)
5. At the **format>** prompt, type **help**

The following text should appear on the screen:

Expecting one of the following: (abbreviations ok):

- 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
show - translate a disk address
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
volname - set 8-character volume name
!<cmd> - execute <cmd>, then return
quit

6. At the **format >** prompt, type **disk**

The **disk** selection lets the administrator choose a disk. Choose the second disk again: **1. c0t2d0**

7. At the **format >** prompt, type **current**

The following text should appear on the screen

```

format> current
Current Disk = c0t2d0: dog
<WDC AC35100L cyl 10670 alt 2 hd 15 sec 63>
/pci@1f,0/ide@d/dad@2,0

```

This shows much the same information as the original selection window. It will be different on different systems.

OPTIONAL

If the second hard drive is not being read properly by Solaris, there is a menu option in the **format** utility to define a hard drive by hand. If the number of cylinders, heads and sectors of the second hard disk are not correct, follow these steps:

1. Look at the hard disk's label for technical information. If the disk does not have a label on the top, visit the manufacturer's website and find the following information for your drive:
 - Number of cylinders
 - Number of heads
 - Data sectors per track
 - Speed in RPM
2. Also, find other people that have added the same third-party disk to a SPARC server. You may be able to find this information on the Internet, or by asking other people. Ask them to send a copy of their **/etc/format.dat** file. Search through this file for the disk settings that match your hard drive.
3. When you have the information you need, run the **format** program.
4. At the **format >** prompt, type the word **type**
5. Select the last option #. (**other**)
6. Try setting the number of data cylinders to the hard drive's physical number of cylinders. As noted above, you may need to get this information from the manufacturer's website, by searching the Internet, or by asking other people.
7. Set the alternate number of cylinders to 0
8. Select the default for all the other settings.
9. When asked to name the disk, type in a name for this type of disk, for example, **mydiskdef**
10. When you are returned to the **format >** prompt, type **save**
11. Save the disk information in the **/etc/format.dat** file
12. At the **format >** prompt, type **quit**
13. Type the command **more /etc/format.dat**

14. The disk definition **mydiskdef** should be inside this file.
15. Experiment with different settings until the disk is recognized.
16. Save a copy of the **/etc/format.dat** file before editing by hand.

When the hard drive's settings are correct, follow these steps:

8. At the **format >** prompt, type **partition**
9. At the **partition >** prompt, type **modify**
The following text should appear

Select partitioning base:

- 0. Current partition table (original)**
- 1. All Free Hog**

Choose base (enter number) [0]?

10. Select option **1. All Free Hog**. This means to "hog all the remaining free space."
When the disk's partition table is shown, check slice 2. It should show the total number of cylinders and disk size that matches the hard drive's geometry and capacity. Please note that It is quite normal for a hard drive to have less disk space than specified on the disk's label. Most manufacturers rely on special software to give the user the full disk capacity under Microsoft Windows. Sun Solaris can still use the disk, except that it will have less overall capacity.

You should see a display like the following:

Part	Tag	Flag	Cylinders	Size	Blocks
0	root	wm	0	0	(0/0/0) 0
1	swap	wu	0	0	(0/0/0) 0
2	backup	wu	0 - 10669	4.81GB	(10670/0/0) 10083150
3	unassigned	wm	0	0	(0/0/0) 0
4	unassigned	wm	0	0	(0/0/0) 0
5	unassigned	wm	0	0	(0/0/0) 0
6	usr	wm	0	0	(0/0/0) 0
7	unassigned	wm	0	0	(0/0/0) 0

Do you wish to continue creating a new partition table based on above table[yes]?
Free Hog partition[6]?

11. Type **7** as the "Free Hog Partition."
The Free Hog Partition is used to take up any extra space that the user does not select.
12. *Eventually, you will see the question:*

Enter the size of partition '0' [0b, 0c, 0.00mb, 0.00gb]:

Type in a number and a size unit for partition 0, as discussed immediately below. Set this to about 1/10th of the disk's space.

You can specify the units for the disk slices by using numbers and the following unit labels:

- *b – blocks*
- *c – cylinders*
- *gb – gigabytes*
- *mb – megabytes.*

For example, 2gb = 2 gigabytes, 30mb = 30 megabytes, 100c = 100 cylinders, 5000b = 5000 blocks

After you enter the size and unit information for partition 0, you will be prompted to enter this information for each of the other partitions, one by one. You will not be asked about **slice 2** because it is only a symbolic representation of the hard drive and cannot be changed or specified.

For example if a hard drive has 10 gigabytes of space, you might answer the prompts as follows (starting from partition 0):

```
0 - 1 gb
1 - 1 gb
3 - 1 gb
4 - 1 gb
5 - 1 gb
6 - 1 gb
7 ← free hog will be 4 gigabytes
```

Notice that you are not asked about **slice 7**. It will simply take up any remaining capacity not already chosen.

13. A final confirmation screen should appear, as show below:

Part	Tag	Flag	Cylinders	Size	Blocks
0	root	wm	0 - 2219	1.00GB	(2220/0/0) 2097900
1	swap	wu	2220 - 2436	100.13MB	(217/0/0) 205065
2	backup	wu	0 - 10669	4.81GB	(10670/0/0) 10083150
3	unassigned	wm	2437 - 2653	100.13MB	(217/0/0) 205065
4	unassigned	wm	2654 - 2870	100.13MB	(217/0/0) 205065
5	unassigned	wm	2871 - 3087	100.13MB	(217/0/0) 205065
6	usr	wm	3088 - 3304	100.13MB	(217/0/0) 205065
7	unassigned	wm	3305 - 10669	3.32GB	(7365/0/0) 6959925

Okay to make this the current partition table[yes]?

14. Type **yes** to confirm the partition table.

15. Enter the table name as **mypart**

When asked if the disk should be labeled, type **yes**

16. At the **partition >** prompt, type **print**

The print command will display the partition table again. It does not change anything.

17. At the **partition >** prompt, type **quit**

18. At the **format >** prompt, type **quit**

The second hard drive has been formatted. However, it can not be used yet. In Chapter 14 will discuss how to create and access a file system on this drive.

Key Points to Remember

This chapter described SCSI and IDE hard drives. It is important that the system administrator know how these disk drives work. The type of disk drive selected affects the performance of the SPARC server. For example, if a hard drive has more platters, that drive will most likely have a faster read/write speed than a comparable hard drive with fewer platters.

The disk naming conventions are a bit awkward to work with initially. Over time, the cumbersome **c0t2d0s4** type format will become second nature. Remember to use the **format** command to look at the hard drives on the server. If the system's firmware or the **format** command do not see a hard drive, there is most likely a mechanical problem with the hard drive or the cabling to the drive.

Chapter 14 Using the Mount and Share Commands

Lessons in This Chapter

Lesson 14. 1 Viewing the Current Mount Points.....	14-2
Lesson 14. 2 Mounting a Second Hard Drive.....	14-3
Lesson 14. 3 Mount the 2nd Hard Drive in Read-only Mode.....	14-6
Lesson 14. 4 Creating an Entry In the <code>/etc/vfstab</code> File.....	14-8
Lesson 14. 5 Sharing a Directory.....	14-12
Lesson 14. 6 Using the <code>dfshares</code> Command.....	14-14

Introduction

One obvious function of networking is sharing files and directories. Solaris 9 has several very nice features used to share and mount file systems on Solaris servers, as well as on non-Solaris servers.

This chapter examines the **mount** command and how it works with local file systems and networked file systems. This command is used to access slices on hard drives located inside the workstation. The **mount** command is also used to access shared directories on other servers and workstations in the network.

The **share** command is the counterpart to the **mount** command. It is used to make a directory accessible to other computers on the network. This chapter shows how to share a directory onto a network.

Understanding Mounting

Mounting a hard drive is a foreign concept to most Microsoft Windows users. In Windows, each hard drive or partition is simply assigned a letter (such as **c:**, **d:**, or **f:**). For example, if a second hard drive were added to a computer running Windows 98, it would be accessed as the **d:** drive.

UNIX uses the concept of a *mount point* to access a hard drive. A mount point starts out as just an ordinary directory. Next, the **mount** command is used to associate a hard drive or partition with the target mount directory. Under UNIX, the second hard drive would be accessed under the mount point name (such as `/my2nddrive`) instead of using a drive letter like **d:** in Windows.

Mount points don't just apply to hard drives. They are also used for other file systems such as floppy drives, CDROM drives, DVD drives and networked hard drives. Basically, any type of local or non-local (networked) media is accessed through a designated directory known as a mount point.

When a new hard drive is added to a server, the slices on the new hard drive must be formatted, a new file system must be created on the slice, and the slice must be mounted. In Chapter 13 the reader used the **format** command to create the file system structure. Now it's time to learn how to use the **mount** command to mount a hard drive and view its contents through a mount point.

Lesson 14.1 Viewing the Current Mount Points

In this lesson readers are going to view the current mount points on the test workstation. The current mount points are stored in a text file named `/etc/mnttab`. This is a file that the `mount` command uses to record what file systems have been mounted.

1. Log in as the root user.
2. Open a Terminal window.
To open a Terminal window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Tools menu item, then left click on the Terminal icon.
3. Type the command `mount`
This command shows the currently mounted partitions. Don't worry about what everything means right now.
4. Type the command `cat /etc/mnttab`
This command shows the currently mounted partitions. The `mount` command uses the `/etc/mnttab` file to keep track of the mounted partitions. Don't worry about what everything means right now.
5. Type the command `clear`
The `clear` command removes all text from the display.
6. Type the command `mount | grep /slice3`
This command only displays mount information for `/slice3` because this `grep` command displays only lines with `/slice3` in the text. (Your system might not have a mount point called `/slice3`).
7. Type the command `cat /etc/vfstab`
The `/etc/vfstab` file is used to save the mount options between reboots. This file will be described in detail later in the chapter.

The following example shows part of a typical display from the `mount` command (a heading has been added and the fields of the display have been lined up):

Mount Point	Status	Disk Slice or Device	Options
<i>/</i>	on	<i>/dev/dsk/c0t0d0s0</i>	read/write/setuid
<i>/proc</i>	on	<i>/proc</i>	read/write/setuid
<i>/etc/mnttab</i>	on	<i>mnttab</i>	read/write/setuid
<i>/dev/fd</i>	on	<i>fd</i>	read/write/setuid
<i>/var/run</i>	on	<i>swap</i>	read/write/setuid
<i>/tmp</i>	on	<i>swap</i>	read/write/setuid/xattr
<i>/slice3</i>	on	<i>/dev/dsk/c0t0d0s3</i>	read/write/setuid

Take a look at the last line. As can be seen from the example, every device or disk slice (such as `/dev/dsk/c0t0d0s0`) has what is known as a *mount point*. A mount point is a directory name that is used to access a disk or disk slice. Here, the directory is `/slice3`.

There are some oddball mount points like `swap`, `mnttab` and `/proc` that are used by the Solaris operating system internally for its own housekeeping operations. For now, just concentrate on the mount points that refer to the local hard drives and local slices (they are shown in italics in the example). The first of these mount points is the root directory (`/`). The second is `/dev/fd`, which is discussed below. The third is `/slice3`.

Take special note of the `/dev/fd` mount point. This is used to access the floppy drive. Because a floppy drive is a removable device, its mount operations are very different than with a hard drive mount point. Unlike a hard drive's file system, a floppy drive's file system can be removed or changed without the operating system knowing that the change has occurred (such as when a new diskette is inserted in the floppy drive). A CDROM or DVD disk is also very different from a floppy drive, because the CDROM or DVD drive notifies the operating system when a CDROM or DVD has been added, removed or changed. A floppy drive does not actively notify the operating system that a diskette has been added or removed. The operating system only finds out that a floppy disk was changed or removed the next time it tries to access that file system.

The **newfs** command is used to create a new file system. The **newfs** command is similar to Microsoft's **format** command. This command can only be run by the root user. The only exception is if an ordinary user needs to format a floppy diskette. The **newfs** command supports the following options:

```
newfs <options> /dev/rdisk/<disk_logical_device_name>  
  
<options>
```

-v Verbose output, gives more detailed information on the **newfs** command and its output.

For example:

```
newfs /dev/rdisk/c0t2d0s0
```

is a very common way of creating a new file system on the second hard drive.

Lesson 14.2 Mounting a Second Hard Drive

In this lesson, readers will access the first slice of the second hard drive, using the **mount** command. Until a directory is created and mounted, the second hard drive file system is inaccessible and it is not possible to read or write files to the second hard drive in the usual way.

The **mount** command is located in the **/usr/sbin** directory. This command can be used by ordinary users if provision has been made for that. Most of the time, the root user is the only one that uses the **mount** command.

WARNING

This lesson uses the **newfs** command. This command will destroy any data on the slice on which it is creating a new file system. If you are unfamiliar with the Solaris disk device naming conventions (example: **/dev/dsk/c0t3d0s4**) re-read the "Solaris Disk Naming Conventions" section in Chapter 13.

When the **newfs** command is used, it will destroy the file system on the second hard drive. Copy any critical information from the second hard drive to another source before running the **newfs** command.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command **format**

You will see a display like
Searching for disks...done

AVAILABLE DISK SELECTIONS:

- 0.** **c0t0d0** <ST315320A cyl 29649 alt 2 hd 16 sec 63>
/pci@1f,0/ide@d/dad@0,0
- 1.** **c0t2d0** <ST315320A cyl 29649 alt 2 hd 16 sec 63>
/pci@1f,0/ide@d/dad@2,0

Specify disk (enter its number):

*Notice that the logical device name of the second hard drive is **c0t2d0** (the first hard drive is usually **c0t0d0**). Logical device names can differ, depending on your test workstation's model number and setup. Write down the*

logical device name of your second hard drive. Use this name in this lesson, and in all subsequent lessons in this book, as well.

4. At the prompt, type **1**.
5. At the **format >** prompt, type **quit** to leave the format utility without performing any functions.
6. Before proceeding, make sure that you understand Solaris logical device names. If you have any concerns, re-read Chapter 13.
Make sure that you have backed up any critical data from the second hard drive.
7. After the second hard drive is backed up, type the command **newfs /dev/dsk/c0t2d0s0** (or use the logical device name you noted in step 3).
*Remember that the logical device name **c0t2d0s0** might be different, depending on the test workstation's model and setup. If it is, substitute that name for **c0t2d0s0** in all steps below!*
8. When asked to continue, answer **y** or **yes**
*The command **newfs** creates a new file system on the first slice of the second disk drive.*
9. Type the command **mkdir /2nddrive**
*The **mkdir** command creates a directory named **/2nddrive**.*
10. Type the command **mount /dev/dsk/c0t2d0s0 /2nddrive**
*This mounts the second hard drive under the directory **/2nddrive**.*
11. Type the command **cd /2nddrive**
*This changes the current directory to the **/2nddrive** directory.*
12. Type the command **touch 2nddiskfile**
*The **touch** command creates an empty file named **2nddiskfile** on the second hard drive.*
13. Type the command **cd /**
This takes you out of the directory you are about to unmount. Solaris does not allow you to unmount the current directory.
14. Type the command **umount /2nddrive**
This unmounts the second hard drive
15. Type the command **ls /2nddrive**
*Notice that the file **2nddiskfile** seems to be missing? It is on the second hard drive, but it is not visible because that directory has been unmounted and is not currently being used as a mount point.*
16. Type the command **mkdir /2nddisk**
*This creates a directory named **/2nddisk***
17. Type the command **mount /dev/dsk/c0t2d0s0 /2nddisk**
18. Type the command **cd /2nddisk**
19. Type the command **ls -l**

```
2nddiskfile  
lost+found
```

*This listing shows the contents of the second drive under a new mount point (**/2nddisk** instead of **/2nddrive**). Note that only the mount point name has changed, not the files. The directory **lost+found** is a directory that is always created on a new UFS file system. This directory is used when file fragments are recovered.*

How to Unmount a Busy or Jammed File System

There are rare times when a mounted file system will not unmount properly. This can happen if a user or application is using the mount directory. It is usually easy to find the application or user and properly shut down the application or ask the user to leave the mount directory. However, sometimes, a program becomes hung or a stale user account keeps a mounted directory jammed open.

To unmount a busy or stuck mounted file system, follow these steps:

1. Type the command **fuser -cu /mountpoint**(where **/mountpoint** is the name of the directory that is jammed). This shows the users and processes that are using it. Try to gracefully end these processes and to get anyone who is using the jammed directory to log off.
2. If the previous step, doesn't work, warn users and then type the command **umount -f /mountpoint**
This will forcibly unmount a file system.

Accessing Removable Media

As mentioned earlier, using removable media, such as CDROMs and floppy disks, provides a unique challenge to the Solaris operating system. These disks can be inserted or removed, resulting in a brand new file system, directories and files being introduced to the system. This section will describe in greater detail how Solaris 9 handles the unique challenges of removable media.

Managing the floppy drive is particularly difficult, because a diskette can be removed or changed without the operating system even knowing that the change occurred. Because removable devices cause such a unique situation, Solaris has provided a special volume management program to deal with CDROM drives, floppy drives and other removable media. This command is known as the **vold** command. Its full path is:

/usr/sbin/vold

The **vold** program is always running in the background, constantly trying to detect the presence or removal of removable media. It will detect the presence of a new CDROM disk because the CDROM drive sends an open/close message to the operating system. However, when a system administrator wants to remove a CDROM or DVD, the command **eject cdrom** should still be typed on the command line. If a CDROM or DVD is removed without this command, the operating system might not recognize that a CDROM has been removed or changed. Most SPARC-based systems will not open the CDROM door unless the **eject cdrom** command is typed.

To access a CDROM after it is inserted, type **cd /cdrom/cdrom0**

When a floppy disk is inserted in an empty floppy drive, or when one floppy is removed and another one is inserted, follow these steps:

1. Type **volcheck** to let operating system know that a new floppy has been inserted.
2. Type **cd /floppy/floppy0**

There are some situations, such as a publicly accessible workstation or kiosk display, where a system should not allow ordinary users the ability to access a CDROM drive or a floppy drive.

To disable the CDROM and floppy drive, type the command:

/etc/init.d/volmgt stop Stops the **vold** daemon and prevents CDROM/floppy access.

To re-enable the floppy and CDROM drive, type the command:

/etc/init.d/volmgt start Restarts the **vold** daemon and authorizes CDROM/floppy access.

Some advanced utilities require accessing a CDROM or floppy in *raw mode* (*character mode*). To do this, use the following directories:

```
/vol/dev/aliases/cdrom0
/vol/dev/aliases/floppy0
```

As we saw in the previous lesson, the second hard drive was not accessible until an ordinary directory was created on it. At the start of this lesson, we created a directory called **/2nddrive**. This allowed the root user to create a file named **2nddiskfile** on the second hard drive.

Anytime a file needs to be written to a slice, a mount point (such as **/2nddrive**) must be created and mounted. There are some exceptions, but for now, just understand that most read/write operations to a file system require a mount point.

There is a special directory the **/var/run** directory. This is a new directory in Solaris 9. It is a new **tempfs** file system that is used to hold temporary files. Do not touch or work with this directory!

Special Mount Options

A partition can be mounted with special options. These options restrict how the file system can be accessed. The **mount** command uses this format:

```
mount -o < options > /device/to/mount /mountdirectory
```

<options>

ro	The read-only option. Files can be read and copied from the file system, but executable files can not be run.
rw	The read/write option. Files can be read and written to the mounted file system.
suid/nosuid	Enables/disables execution of setuid programs on the mounted hard drive. This is for security purposes so that a hacker can't execute a file with root permission.
nologging/logging	Enables/disables keeping track of the actual reads and writes to a file system. If a read/write operation is not performed successfully or the connection is broken during the read/write, the file system will not become corrupted. Enabling logging increases the time needed to read and write files to a hard drive.
largefiles/nolargefiles	Enables/disables creation of files greater than 2 GB in size. Some older versions of Solaris can not handle large files.

Lesson 14.3 Mount the Second Hard Drive in Read-only Mode

In this lesson the readers will gain access to the second hard drive. The mount point **/2nddisk** will be mounted as read only. This will prevent creation of a file or directory on the second hard drive. In essence, the second hard drive's content will be read only.

This lesson assumes that the logical device name for the second hard drive first slice is **/dev/dsk/c0t2d0s0**. If you recorded a different logical device name for your drive in Lesson 14.2, use that name in this lesson and all subsequent lessons.

1. Log in as the root user.
2. Open a Terminal window

3. Type the command **mkdir /2nddrive**
This command creates a directory named /2nddrive
4. Type the command **mount -o ro /dev/dsk/c0t2d0s0 /2nddrive**
This mounts the second drive on the /2nddrive directory.
5. Type this command **cd /2nddrive**
This changes to the /2nddrive directory, to access the second hard drive.
6. Type the command **touch anewfile**
The command tries to create a file called anewfile. This should not work, because the second hard drive was mounted as a read only hard drive.
7. Type the command **cd /**
This command switches out of /2nddrive, so that it can be unmounted.
8. Type the command **umount /2nddrive**
This commands unmounts the second hard drive.
9. Type the command **mount /dev/dsk/c0t2d0s0 /2nddrive**
This command mounts the second hard drive again, but with the default read/write permissions.
10. Type the command **touch /2nddrive/anewfile**
This command works, because the second hard drive was mounted with the default read/write permissions.

The /etc/vfstab File

In the previous lessons the **mount** command was typed at the command line. This created a mount point, but the mount point disappears when the system is shut down or rebooted.

Sun Solaris provides a way to keep mount settings and mount a slice automatically when the system starts. The text file **/etc/vfstab** contains automatic mounting options. If this file becomes damaged or the proper syntax is not followed in it, the server could become severely damaged and not start properly.

Never work with the **/etc/vfstab** file without creating a backup copy of the file. This file is absolutely critical to the operation of a server!

Figure 14.1 shows what a typical **/etc/vfstab** file looks like.

(1)	(2)	(3)	(4)	(5)	(6)	(7)
Device to Mount	Device to fsck	Mount Point	FS Type	fsck Pass	Mount at Boot	Mount Options

Figure 14.1 Fields in an /etc/vfstab file

The **/etc/vfstab** file has seven fields. They are:

Field	Field Title	Description
1.	Device to Mount	What disk and slice should be automatically mount when the system starts.
2.	Device to fsck	The fsck program checks and fixes a file system. This program is very similar to Microsoft's Scandisk . This field tells the fsck command what file system to check. The fifth field (5) works in conjunction with the second field in that it tells the fsck command how to check the file system. The key point here is that the second field tells fsck what to check.
3.	Mount Point	Where to mount the slice or partition.

- | | | |
|----|---------------|---|
| 4. | FS Type | Some of the operating system's mounted file systems are special file system types (tmpfs-memory based, procfs-process representation, swapfs-swap partition, fdfs-floppy device) and require these special mounting options. |
| 5. | fsck Pass | Pass Indicates if the file system should be checked with the fsck command when the system starts. A zero (0) or dash (-) indicates that the file system should not be checked on startup. Any number 1 or higher indicates it should be checked on startup. |
| 6. | Mount at Boot | Answers the question "Should this file system be mounted when the system is started?" If the file system has a "no" in this field it still can be mounted if the root user types mount /mountpoint . If the root user types the command mountall all file systems with a "yes" in this field will be mounted (provided they are not already mounted.) Also, if the root user types the command mountall -l this will mount all local file systems (provided they are not already mounted). |
| 7. | Mount Options | These are the same mount options possible as with the command line mount -o < options > . |

Lesson 14. 4 Creating an Entry in the /etc/vfstab File

The **/etc/vfstab** file is read during the boot process. If this file becomes damaged, critical file systems like the root (**/**) and swap (**slice 1**) partitions might not be mounted. The server will come to a halt.

WARNING

The **/etc/vfstab** file is a very critical file for the system. Always make a backup copy of **/etc/vfstab** before you work with it! This file is used to determine what file systems are mounted when the system boots. If this file becomes corrupted, the Solaris 9 Operating System will not boot properly. The **/etc/vfstab** file is very critical, be careful with this file.

1. Log in as the root user.
2. Open a Terminal window.
To open a Terminal window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Tools menu item, then left click on the Terminal icon.
3. Type the command **mkdir /mymount**
4. Type the command **cp /etc/vfstab /etc/vfstab.old**
*The command **cp /etc/vfstab /etc/vfstab.old** saves a copy of the **vfstab** file to a backup file. If something really bad happens, type the command **cp /etc/vfstab.old /etc/vfstab**. This will restore the last good **vfstab** to its original condition.*
5. Type the command **vi /etc/vfstab**
*The next four steps describe how to edit the **/etc/vfstab** file in the safest way possible. If you make a mistake, type the **vi** command **:q!** and start again from the step after this one.*
6. Press the down arrow key several times until the cursor is at the last line.
*If the arrow keys produce funny letters instead of moving the cursor, you have a problem with your **TERM** variable. If this happens, type the command **:q!** to exit the **vi** editor. Then try setting the **TERM** variable to **TERM=vt100** with the command*
TERM=vt100 ; export TERM

if the vi editor still causes problems, try changing the variable to
`TERM=sun ; export TERM`

Also note: Microsoft's HyperTerminal program is suspected of having a software flaw that makes it incompatible with Solaris. There are other free terminal emulators for Microsoft Windows on the Internet, such as **Tera Term Pro** and **PuTTYtel**, that do not have the same problems.

7. Press the capital letter O (not the number zero, the letter "O")
8. Type in the following line. Press TAB key between words:

```
/dev/dsk/c0t2d0s0 /dev/rdisk/c0t2d0s0 /mymount ufs 1 no -
```

!!! ASK A SENIOR LEVEL ADMINISTRATOR TO CHECK THE FILE BEFORE YOU DO ANYTHING MORE !!!

9. Press the ESC key.
10. Type the command `wq!`
11. Type the command `mount /mymount`

The `mount` command reads the `/etc/vfstab` file for the device to mount under the `/mymount` directory. If for some reason the `/mymount` directory does not mount properly, check the `/etc/vfstab` file for errors. If necessary, replace the new `/etc/vfstab` file with the original file with the command `cp /etc/vfstab.old /etc/vfstab` and start step 5 again.

Do not go to the next step, and do not otherwise reboot the workstation, until the `/etc/vfstab` file is in perfect working order!

12. Type the command `init 6`

This reboots the workstation. The new version of `/etc/vfstab` will be read in.

The `/usr/sbin/umountall` Command

The command `umountall -l` unmounts all local files in the `/etc/mnttab` file. This is a dangerous command to use because it can unmount file systems that the operating system needs to function. Also some third party database programs can become unstable if they can not access a mount point.

DO NOT TO USE THIS COMMAND EXCEPT IN EMERGENCIES!!

Mounting Different Media

There are occasions when special file systems need to be mounted

- **hsfs** The CDROM format
- **pcfs** The floppy diskette format
- **nfs** A file system mounted across a network

The `mount` command, when used with special types of file systems supports the following options:

```
mount -F <special file system> /mount/device /mount/point
```

```
mount -o ro -F hsfs /dev/dsk/c0t3d0s0 /MyCdromMountPoint
```

This command mounts a CDROM drive.

```
mount -F pcfs /dev/diskette /MyFloppyMountPoint
```

This command mounts a floppy drive.

If you do not know what type of file system you are dealing with, try typing the command:

```
fstyp /dev/rdisk/c0t0d0s0
```

This command will return the file system type present.

To determine how a file system is currently mounted (what file system type was used to **mount** a hard drive or CDROM) check the **/etc/vfstab** file and the **/etc/mnttab** file.

If the file system type is not specified with the **-F** option, the **mount** command will try and guess what file system is present. There are three text files that it can read from:

/etc/vfstab If the **mount** command finds the mount point or device here, it will read the fourth column (FS type) and mount the device with that file system type.

/etc/dfs/fstypes If the **mount** command believes that it has found a remote file system, it will check this file to try and figure out what the remote system's type.

/etc/default/fs If the **mount** command still can not be determined, the **mount** command will use the **LOCAL=ufs** entry for the default local file system type.

If the **mount** command still can not determine the proper file system type after trying the types shown above, it will give an error message and not mount the unknown file system.

Sharing a Directory on a Network

This section deals with sharing a directory on a network and accessing that shared directory over a network. The directory can be mounted from a client workstation or client server. The files in the shared directory can be read or written to by the client computer (depending on the permissions granted by the server). These files are shared through the DFS (Distributed File Systems). The DFS architecture supports RPC (Remote Procedure Calls) and XDR (eXternal Data Representation). These files can be accessed by any operating system that supports the DFS architecture, such as Linux, Novell NetWare, or Microsoft Windows.

Common Terms

NFS server Any computer that shares its files on the network. An Ultra 5 workstation could be considered an NFS (Network File System) server if it shares its files over the network. The key point is that any computer that shares its files over the network is a server, regardless of its typical function.

NFS client Any computer that connects to a server and reads the files on the server. If an E10K (an E10K is high end server) is reading files from a workstation, the E10K server is technically acting as an NFS client to the workstation that is acting as an NFS server. Understand that it is the *network* function that determines if a computer is a server or a workstation, not its usual role or model name.

Using the **share** Command

When a directory is shared, clients can read and write to the files on the shared directory. The **share** command is used to share a directory and set the server side permissions on that directory.

The **share** command supports the following options:

share <options> <directory>

<options>

- d <description text>** Lets you enter a text description that can be seen by the client
- F <type>** Specifies the type of file system that is being shared. When a client computer connects to the server, the server tells the client what type of file system it is sharing out. If the **-F** option is not used, the first line in the **/etc/dfs/fstypes** file is used, which is NFS.
- o** Sets specific read/write/root access permissions on a directory.
 - ro** Grants read only permission.
 - rw** Grants read and write permission.
 - root=client** Lets the client computer perform root level permissions inside the shared directory.
 - anon = <number >** Sets a specific UID for the client user inside the shared directory. If **anon=-1** is used, access is denied. By default, client users are given the "nobody" UID.

Using Share Access Lists

The **share** command supports the option to set different share levels for specific computers, domains, networks or NIS/NIS+ domains. These access lists support the following convention:

- hostname** A hostname that is recognized by the server.
- @network** Any network name (for example @mynetwork.com) specified in the **/etc/networks** file or any network number (for example 234.18.203) that is in the **/etc/networks** file.
- .domain** Clients from a specific domain.
- netgroup-name** Clients from a NIS/NIS+ netgroup

The next couple of examples show how to use share access lists:

```
share -o rw=client1:client2:client3 /mysharepoint
```

Several clients can be listed with a colon (:) separating the client names.

```
share -o ro=@sun.com:@uswest.com:@cnn.com /mysharepoint
```

Several networks are listed with a colon (:) separating the network names.

```
share -o rw,ro=client1:client2 /mysharepoint
```

The comma (,) is used to separate different types of **-o** options. The colon (:) is used to separate different types of clients (such as hosts, networks and domains).

After the **share** command is run, the nfs daemon needs to be started. To start the NFS server and share a directory on the network, type the command:

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

Lesson 14.5 Sharing a Directory

This lesson can only be performed by a user who has two Sun workstations or two systems that are running Solaris 9. If you only have one Solaris workstation, follow the lesson as closely as possible. Just understand that this lesson *should* be run with two workstations.

In this lesson the readers will create a directory named **/myshare**. The readers will then share the directory on the network. Another workstation will be used to access the shared directory with the **mount** command. Substitute the following symbolic names:

share-workstation The hostname of the workstation that shares the directory.
mount-workstation The hostname of the workstation that mounts the shared directory.

When using the following lesson, substitute the symbolic hostnames of your systems for these names. For example, if there is a command:

```
share share-workstation /mydirectory
```

type the command

```
share sun100 /mydirectory
```

or

```
share <workstation_host_name> /mydirectory
```

 with the appropriate workstation's host name.

1. Log in to the workstation that shares a directory (**share-workstation**) as the root user.
2. Open a Terminal window
3. Type the command **mkdir /mysharedirectory**
This command creates an ordinary directory named /mysharedirectory
4. Type the command **share -F nfs /mysharedirectory**
This command shares the /mysharedirectory onto the network.
5. Type the command **/etc/init.d/nfs.server start**
This command starts the nfs daemon necessary for sharing files. If the nfs.server does not start, add the following line to the /etc/dfs/dfstab file
share -F nfs /mysharedirectory
Now type the command: /etc/init.d/nfs.server start again.
6. Type the command **share**
This command shows what directories are shared on the network.
7. Type the command **cat /etc/dfs/sharetab**
This command shows the contents of the /etc/dfs/sharetab file. That file shows what directories are shared on the network.
8. Type the command **touch /mysharedirectory/<your_first_name>**
This command makes a text file with your first name.
9. On another workstation (**mount-workstation**) log in as the root user.
10. On **mount-workstation** type the command **mkdir /networkshare**
This command creates a directory named /networkshare. Right now, this is just an ordinary directory.
11. On the **mount-workstation** type the command

```
mount share-workstation:/mysharedirectory /networkshare
```

Note: The hostname of the sharing system must be known by this workstation. If the command `ping share-workstation` does not produce a response, the `mount` command will not work.

Type the command `cat /etc/hosts` to see if the client's hostname is in the `/etc/hosts` file.

Edit the `/etc/hosts` file so the client's hostname and IP address are in the `/etc/hosts` file. The `mount` command should then work.

The `mount` command accesses the sharepoint from the NFS server named `/myshare`. If the `mount` command still does not work, try this command:

```
mount 192.168.0.4/mysharedirectory /networkshare
substitute the 192.168.0.4 IP address with the correct IP address of the share-workstation.
```

12. Type the command `ls /networkshare`

The `ls` command should show the contents of `share-workstation`, the text file with your first name.

Using the unshare Command

When a directory no longer needs to be shared, the `unshare` command is used to remove the share point. The `unshare` command supports the following options:

```
unshare <options> <directory>
```

<options>

-F <file system type> This option is used to specify the type of file system being unshared. Some older file systems do need to know what is being unshared, even though they are connected to a share point and have accessed the file system.

<directory> The directory that is shared.

For example:

```
unshare -F nfs /mysharedirectory
unshare /mysharedirectory
```

The shareall and unshareall commands

The `shareall` command is used to share all the files in the `/etc/dfs/dfstab` file. The `/etc/dfs/dfstab` file is used to specify directories that should be shared every time the server starts. Figure 14.3 shows a copy of a typical `/etc/dfs/dfstab` file.

```
# Place share(1M) commands here for automatic execution
# on entering init state 3.
#
# Issue the command '/etc/init.d/nfs.server start' to run the NFS
# daemon processes and the share commands, after adding the very
# first entry to this file.
#
# share [-F fstype] [-o options] [-d "<text>"] <pathname> [resource]
```

```
# .e.g,
# share -F nfs -o rw=engineering -d "home dirs" /export/home2
share -F nfs /install
share -F nfs /install2
```

Figure 14.3 The /etc/dfs/dfstab File

The command **unshareall** reads the `/etc/dfs/sharetab` file. This file has a listing of all the shared directories on the server. The **unshareall** command revokes all the shared directories on the server.

The dfshares Command

The **dfshares** command shows all the directories that are being shared on the network. The **dfshares** command supports the following options:

```
dfshares <options> <host>
```

```
<options>
```

-F The type of file system being shared.

Lesson 14.6 Using the dfshares Command

This lesson uses the **dfshares** command to illustrate what directories and file systems are being shared on the network.

1. Log in as the root user.
2. Open a Terminal window
3. Type the command **dfshares**
This command shows all the shared directories on the server.
4. Type the command **dfshares <share-workstation>**
Enter the name of a client computer accessing a shared resource on this computer.
This command shows what directories the client computer is accessing.

Special Network Mounting Options

The **mount** command is used on the client computer to access the server's shared resources. The **mount** command supports several special mount options:

```
mount <options> server:/sharepoint /mountpoint
```

```
<options>
```

-F <type of file system > Used to specify the type of file system being mounted.

-o < mount options > The mount options on the client side are:

```
rw - read/write
```

```
ro - read only
```

The **mount** command can access several read only sites for redundancy. For example;

```
mount -o ro server1:server2:server3:/share/directory
/mynetworkmountpoint
```

In this example, if the share point **server1:/share/directory** become inaccessible, then the next server in the list (**server2**) is contacted. If its share point **server2:/share/directory** become inaccessible, then the last server on the list is contacted. This only works with read-only directories.

To mount a directory when the NFS client first boots, add a line into the **/etc/vfstab** file.

# device	device	mount	FS	fsck	mount	mount
# to mount	to fsck	point	type	pass	at boot	options
server:/share/directory	-	/mymountdirectory	nfs	-	yes	<options>

Key Points to Remember

A new hard drive needs to be mounted to work. Unfortunately, this concept is very foreign to users of Microsoft Windows. If the second hard drive on the system can not be *seen* by the **mount** command, re-read Chapter 13. A second hard drive needs to be formatted and have a file system added with the **newfs** command.

The **share** command is the counter-part of the **mount** command. The **share** command makes a directory accessible over the network by other systems. If there is a read/write problem between two computers, check both the **mount** and **share** permissions and options.

Another key point to remember is that a SPARC system is a client or a server, depending on its role. An Ultra 5 workstation functions as a server if it is sharing out a directory over the network. Most readers become confused over the term *workstation* or *server* because of the system's name (E-450 Server, E-4500 Server). A server or client is determined by its role, not its marketing title.

Chapter 15 Solaris File Systems

Lessons in This Chapter

Lesson 15.1 Using the <code>prtvtoc</code> Command	15-5
Lesson 15.2 Using the <code>du</code> Command	15-9
Lesson 15.3 Using the <code>df</code> Command	15-9
Lesson 15.4 Using the <code>quot</code> Command	15-10
Lesson 15.5 Using the <code>fsck</code> Command	15-12
Lesson 15.6 Using the <code>newfs</code> Command	15-12

Introduction

This section will examine the different file systems that work on Solaris 9. The three main categories of file systems used with Sun Solaris are: local file systems, distributed file systems and pseudo file systems. Local file systems are on resources local to a workstation, such as the hard drive, CDROM and floppy diskette. Distributed file systems (also known as NFS file systems) are file systems that are distributed or shared over a network . A pseudo file system is a special memory-based file system that is primarily used by the operating system for internal use.

Local File Systems

Local file systems are the typical file systems found within a workstation. These file systems all reside on a hard disk, CDROM, DVD, floppy diskette or other common media found on a workstation.

ufs	Unix File System	A file system on a hard drive.
hsfs	High Sierra File System	A CDROM file system.
pcfs	Personal Computer File System	A floppy diskette file system, read by Microsoft Windows.
udf	Universal Disk Format	Used on special CDROMs and DVDs with optical storage devices

Distributed File Systems

Distributed file systems reside on servers across a network. As the name implies, they are “distributed” to many different computers.

Networked File Systems

NFS	Networked File System	Any file system that is accessed over a network. For example, if a server named Johnson had a shared directory named /bringme . A workstation connects to the Johnson server. The file system in the /bringme directory is then considered an NFS file system.
------------	-----------------------	--

Pseudo File Systems

These file systems are not used for typical read/write file operations. They are primarily used internally by the operating system for housekeeping. Most ordinary users can not even access these file systems. These systems are optimized for different type of read/write operations such as memory swapping, and system processes.

swaps Swap File System. The swap file system is used when programs and the operating system demand more memory than is installed on the server. The operating system uses part of the hard drive as if it were memory. Most of the time the swap partition is slice (1) but it can also be a file on the hard drive.

fdfs File Descriptor File System. This has the explicit names of open files.
/dev/fd/0
/dev/fd/1
/dev/fd/2

procfs Process File System. This is under the /**proc** directory. All active processes are kept under this directory.

tmpfs Temporary File System. This is a special file system that allows any user the ability to copy files directly into memory, just like to an ordinary directory. The disadvantage to the tmpfs file system is that the files are destroyed whenever the system is shut down or crashes. When you copy a file into the /**tmp** directory, that file only resides in memory. When the system power goes, the file is gone.

UFS Disk Structure and the File System Structure

The easiest way to describe the UFS file system is to create a rather lengthy story example.

Let's pretend you are an engineer working at Sun Microsystems. You are responsible for creating a Solaris file system. A brand new unformatted hard drive is given to you. You are also given a special device, a hard drive data terminal, that can write raw sector data to the hard drive's surface directly. Understand that this is a fictitious example used to describe the Solaris file system. There is not such thing as a "hard drive data terminal." Read this example as it explains the structure of the UFS file system in Solaris. The **format** command and the **newfs** command create these UFS file system components.

The first thing you need to do is to format the hard drive into cylinders. Remember that a cylinder is nothing more than a collection of tracks on multiple platters. Also remember that a track is the invisible path that a disk head makes around a platter when the platter spins under the disk head.

So you now create cylinders and sectors on the raw hard drive. Remember, each cylinder is broken down into small fragments know as sectors. So now you have a hard drive that is broken down into cylinders, tracks and sectors. Great! now what?

The first thing you need to do is to create a disk label. The disk label describes several key points of information. One of these pieces of information is the disk's geometry (number of sectors on the hard drive, the number of cylinders on the hard drive, and the size of the disk). This is necessary so the operating system knows what kind of disk it is working with. The operating system needs to know that the disk has 23843 cylinders and 183,253 sectors and that the hard drive can hold 12 GB of data.

You save this disk label on the first sector of the hard drive. Some hardware engineers come by your desk and inform you that the Solaris hardware looks at the second sector for a program named the **bootblk**. The **bootblk** loads some other programs into memory to start the boot process of the Solaris operating system. No problem! You copy this **bootblk** program onto the second sector.

Ok, now you have a hard drive with a disk label written to the first sector and the **bootblk** program written to the second sector. Now it's time to create some partitions (*partition* and *slice* mean the same thing here) to save the data on. These partitions are defined by a starting and ending cylinder.

For example:

First partition – starts at cylinder 0 – ends at cylinder 23

Second partition – starts at cylinder 24 – ends at cylinder 200

Third partition – starts at cylinder 201 – ends at cylinder 523

Let's say that you have noticed that the disk's label (also known as the *Volume Table of Contents* or *VTOC*) has some space inside it. So you now decide to save the partition table inside the disk label.

Just as a recap. You have a hard drive. The first sector (512 bytes) has a disk label. The disk label has information about the disk geometry (number of cylinders, number of sectors and size of the disk). The disk label also holds the partition table. The partition table defines the partition number and its start and stop cylinders). The second sector contains the **bootblk** program needed to start the boot process.

Let's say that you just learned that by creating clusters of 16 cylinders, you can improve the hard drive's read/write speed. In other words, if an operating system tried to write to a partition by referencing a cylinder and sector, it would be very slow because it would have to remember a rather large numbers of cylinders and sectors for a 9 GB partition. A cylinder group helps to ease the maintenance necessary on the operating system.

Ok, so now you want to use these cylinder groups, but, um.. how can you create cylinder groups, and where will you put them? The first Cylinder Group will start on the third sector because the first two sectors have the disk label and **bootblk** written to them. Every other cylinder group will written on the hard drive in sequential order, until the hard drive is filled. So how do you keep the housekeeping information for the cylinder groups? Hey! Why not create a "Cylinder Group Block " in the first cylinder of the 16 cylinders that makes up a cylinder group?

Every cylinder group holds the following four components:

- | | |
|--------------------------------------|---|
| Inode Table | Every file is saved under a file name and an inode. The file name is created by a human being and is easy to remember, like myresume . But computers don't like to work with names, so every file has a corresponding inode. The inode is given a number, known as the <i>inode number</i> . An inode stores information about a file. For example, inode # 230623 contains information about a file, such as "File was created by John Anderson; created on August 13, 2002; contents of the file are in data blocks 203, 204, 206; permission R/W/X for everyone...) |
| Cylinder Group Block | This holds all the housekeeping information needed for this cylinder group (number of inodes inside the cylinder group, number of directories inside the cylinder group, number of data blocks inside the cylinder group, where the free data blocks are, where the free inodes are, where the free file fragments inside the cylinder group are, a map of the free data blocks, and a map of the free and used inodes) |
| Data Blocks | This is where the file's contents are stored. When a file is created, its name, creation date and permissions are stored in an inode, and its contents are stored in data blocks. |
| Backup Copy of the SuperBlock | The Superblock is described below. For now, know that the Superblock, or a copy of the Superblock, exists in every cylinder group. |

Great, so now when the Solaris Operating system wants to read or write to the disk, it will contact the cylinder group block for information on what inodes are inside the cylinder group block are free, what data blocks are available, and so forth. That is much easier than trying to figure out what data blocks and inodes are free inside a 16 gigabyte partition.

Ok, so now we have a hard drive with a disk label and the **bootblk** program. Every 16 cylinders are grouped together in what is known as a “cylinder group.” Inside each cylinder group is a cylinder group block, an inode table and the data blocks. So now what?

Well, according to Sun Microsystems, the Solaris operating system needs to have information about each individual slice (remember, the terms *slice* and *partition* are the same). This information is stored in a convenient location known as a *SuperBlock*. The Superblock needs to hold cumulative information about the partition (total number of data blocks on the hard drive, the size of the data blocks on the hard drive, and so forth). Yes, some of the Superblock’s information is repetitive to the cylinder group and disk label, but the Solaris operating system wants to have this information in one clean location.

The Superblock must hold the following pieces of information:

1. The total number of cylinder groups in the slice.
2. The size of the data blocks and fragment blocks in the slice.
A fragment block is used to hold the small fragment of a file that is too small to fit into a data block. The fragment block can hold a maximum of two file fragments.
3. The name of the mount point used to access the slice.
4. A copy of the disk’s label.
5. The file system’s status (clean, active, stable, logging, unknown).
6. The total number of data blocks on the hard drive.

There’s a problem here. Where should you put the superblock? You might decide to put the Superblock inside the first cylinder group and also put a backup copy of the Superblock in the all other cylinder groups. So now the first cylinder group would look like this:

Superblock
Backup Copy of the Superblock
Cylinder Group Block
Inode Table
Data Blocks and Fragment Blocks

Let’s say that you are worried the first cylinder group could become corrupted. So now you are going to put a backup copy of the Superblock so in each and every cylinder group. Except for the first Superblock, all the other cylinder groups would look like this:

Backup Copy of the Superblock
Cylinder Group Block
Inode Table
Data Blocks and Fragment Blocks

Congratulations. You have just created a Sun Solaris UFS file system! The first sector (512 bytes) holds the disk label. The disk label contains information about the hard drive’s geometry (cylinders, sectors, size) and partition table (contains the partition numbers and each partition’s start and end cylinders, along with some extra information covered in detail later). All of the cylinders on the disk are grouped together into cylinder groups that contain 16 cylinders. The first cylinder group holds the Superblock, a backup copy of the superblock, a cylinder group block, an inode table, and data blocks. All other cylinder groups hold the same components, except that they don’t hold the original Superblock, just copies of the Superblock.

Lesson 15.1 Using the `prtvtoc` Command

Use the `prtvtoc` command to view the disk's label.

1. Log in as the root user.
2. Open a Terminal window
To open a Terminal window, right click anywhere in unoccupied desktop space, left click on the Tools menu item, right click on the Terminal icon.
3. Type the command `prtvtoc /dev/rdisk/c0t0d0s2`
The `prtvtoc` command shows the disk's label.

```
* /dev/rdisk/c0t0d0s2 partition map
*
* Dimensions:
* 512 bytes/sector
* 63 sectors/track
* 15 tracks/cylinder
* 945 sectors/cylinder
* 10672 cylinders
* 10670 accessible cylinders
*
* Flags:
* 1: unmountable
* 10: read-only
*
* Unallocated space:
*   First Sector Last
*   Sector Count Sector
* 1638630 1890 1640519
*
*           First Sector Last
* Partition Tag Flags Sector Count Sector Mount Directory
0 2 00 2096010 7987140 10083149 /
1 3 01 0 1638630 1638629
2 5 00 0 10083150 10083149
3 0 00 1993950 102060 2096009 /slice3
4 0 00 1891890 102060 1993949 /slice4
5 0 00 1789830 102060 1891889 /slice5
6 0 00 1687770 102060 1789829 /slice6
7 0 00 1640520 47250 1687769 /extra
```

File System Components in Detail

This next chapter section will describe some common components of a UFS file system in detail. These are the same components that were described in the Sun engineer example.

The SuperBlock

The first cylinder group holds the master copy of the SuperBlock. The SuperBlock has detailed information about the structure of the partition, such as the size of data blocks and fragment blocks, if the file system was cleanly unmounted, and the names of mount points. Some of this information is created from the disk label and is repetitive to the disk label.

The exact information kept in the superblock is:

- File system status. This answers the questions “Was the file system mounted or unmounted cleanly? When was each file written to and read from the file system? Were the operations completed successfully?”

The file system status has the following flags

- Clean
 - Stable
 - Active
 - Unknown
 - Login
- Number of datablocks and fragment blocks inside the partition
 - Number of cylinder groups inside the partition
 - Size of the data blocks and fragment blocks. This size can be changed by senior level system administrators for improved file system performance. The default size is 8 KB data blocks.
 - Description of the hardware (cylinders, sectors, and disk size; taken from the disk label)
 - Name of the mount points of the slice. If a file system is mounted (**/mymountpoint**) that name is saved in the Superblock that corresponds to this slice.
 - Backup SuperBlock: A direct copy of the Superblock.

Cylinder Group Block

A cylinder group block is basically a housekeeping block for a cylinder group. It contains the following information for the cylinder group:

- Number of directories
- Number of inodes
- Number of data blocks
- Number of free blocks, free inodes and free fragment blocks
- Map of the free blocks
- Map of the used and free inodes

The Inode Table

Each file or directory (a directory is nothing more than a file that only contains the names of other files or directories) has a name, such as **myfile** and an inode number, such as “20364.”

The Solaris Operating system would have a hard time working with file names like **myfile**, so it creates an inode file inside the inode table. The inode file’s name is its inode number. An inode table saves these inode files.

An inode holds all the information relating to a file, such as:

- UID – userID taken from the **/etc/passwd** file

- GID – taken from the `/etc/passwd` file
- Date the file was created, modified and accessed
- File access modes – such as `rw-r-x-r-x` (remember the `chmod` and `ls -l` commands?)
- Size of the file
- Link count (a count of the hard links or names of the file)
- Data block count – total number of data blocks that hold the file’s contents.
- Location of the first 12 data blocks of the file contents. (Remember, the inode DOES NOT CONTAIN FILE CONTENTS!!! It only contains the location of the data blocks. The data blocks and fragment blocks hold the actual data.) The number and type of pointers vary with the size of the file. The section immediately below discusses how pointers work with inodes.
- Shadow inode

As mentioned above, an inode contains information on the first 12 data blocks of a file. This means that if a file is less than 96 KB in size it can be referenced by an inode. But what happens if a file is larger than 96 KB?

If a file is larger than 96 KB, the first 11 data block pointers point to the data blocks. Pointer #12 points to a special data block known as a *single indirect pointer block*. This pointer can give the addresses of 2048 other data blocks. The single indirect pointer block does not contain any extra file data. It only points to other “real data blocks.” If a file is less than 16 MB in size it will have a single indirect pointer that references the other 2048 data blocks necessary to save the file’s contents.

If a file is between 16 MB and 32 GB it will be referenced by two second level indirect pointers. The 12th and 13th pointers inside the inode will point to the two second level indirect pointers. These second level indirect pointers reference up to 2048 first level indirect pointers. Each of the first level indirect pointers references 2048 data blocks or fragment blocks.

If a file is larger than 32 GB, but less than 70 TB (terabytes), it will use three triple level indirect pointers. The first 11 pointers point to data blocks. The 12th, 13th and 14th pointers point to three triple indirect pointers. These triple indirect pointers point to 2048 second level pointers. These second level indirect pointers reference 2048 single level indirect pointers, and the single indirect pointers reference 2048 data blocks each. (It is unlikely that a 70 TB file will be created within the next couple of years!)

The list above mentioned *shadow inodes*. When a file has an Access Control List (ACL) associated with it, the shadow inode contains the special security information needed to make the file extra secure. The shadow inode points to a single data block. This data block contains the names of the groups and users, and their restricted rights to the file.

Data Blocks and Fragment Blocks

Data blocks hold the actual file information. By default, a data block is 8 KB, or 8192 bytes, in size. When a file is saved, its contents are saved in data blocks, and information such as the file’s name, creation date, owner and size, is stored in the inode table.

So what are fragment blocks? These blocks hold the last little pieces of files that don't fill up an entire data block.

For example, say that a user creates a file that is 8193 bytes long. If a file system only has data blocks, it would take two data blocks to store the file's contents. The first data block would hold the first 8192 bytes of the file, and the second block would hold just 1 byte of information

8193 bytes = 8192 bytes (first block) + 1 byte (second block)

Obviously, with thousands of files, a great deal of disk space would be wasted by storing only a small fragment of a file in an entire data block. Why waste a data block for just a small fragment?

To solve this problem, the Solaris file system uses what are known as *fragment blocks*. A fragment block is nothing more than a data block that has been converted for special duty. A fragment block holds the last little piece of two files that would otherwise have been stored in two separate data blocks.

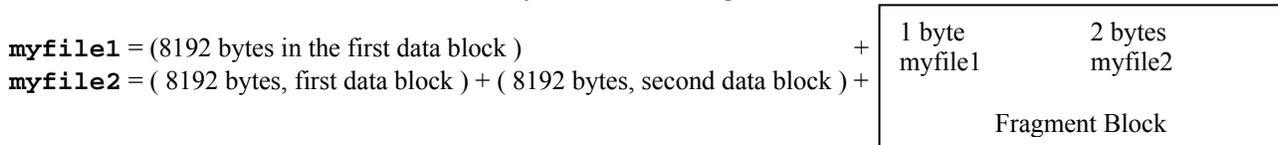
For example, say that two files are created, **myfile1** and **myfile2**. The sizes of the files are:

myfile1 = 8193 bytes
myfile2 = 16386 bytes

With a file system that only has data blocks:

myfile1 = (8192 bytes saved in the first data block) + (1 byte saved in the second data block)
myfile2 = (8192 bytes, first data block) + (8192 bytes, second data block) + (2 bytes, third data block)

Now, consider how the files are stored in a file system that uses fragment blocks:



As shown above, the last little pieces of **myfile1** and **myfile2** are saved in a fragment block. Two complete data blocks don't have to be used to save these small data fragments. Remember that a fragment block can only hold the fragments of two different files.

If one of the files grows to the point where it needs another data block to hold its information, a new data block will be allocated to the larger file, and any extra storage will be put into a new fragment block.

Monitoring a File System

Occasionally, a file system may start to become too full with files. The **du** command is used to see the capacity of a particular directory. It is important that a system administrator can monitor the file system's capacity. The commands in this section are useful ways to do this. The **du** command supports the following options:

du <options> <directory or file>

<options>

- a** Recursively shows all files and directories under a directory
- s** Displays the output in blocks. Each block is 512 bytes, or ½ kilobyte in size.
- k** Displays the output in kilobytes .
- h** Displays the output in a human readable format (kilobytes, megabytes, gigabytes)

Lesson 15.2 Using the **du** Command

The **du** command shows the size, in blocks or kilobytes, of a directory, its subdirectories, and the files in that directory and its subdirectories. This lesson shows the various **du** options.

1. Log in as the root user.
2. Open a Terminal window
3. Type the command **du -sk /etc**
This command gives the total size of the **/etc** directory in kilobytes.
4. Type the command **du -h /etc**
This command gives the total size of the **/etc** directory in human readable format (kilobytes, megabytes, gigabytes)
5. Type the command **du -k /etc**
This command gives the total size of every directory and subdirectory in the **/etc** directory in kilobytes.
6. *Type the command **du -s /etc***
This command gives the total size of the **/etc** directory in blocks (blocks is the default). A block is ½ KB. Notice that the number of blocks is two times the number of kilobytes. Do not confuse this type of block with a Data Block that is 8 KB in size. A Data Block is an internal structure to the file system.
7. *Type the command **du -ak /etc***
This command shows all files and directories in the **/etc** directory in kilobytes.

The **du** Command

This command shows the total amount of blocks or kilobytes of data used by an entire file system. This is not the same thing as the **du** command. The **du** command that is designed more to show the size of a directory. The **df** command supports the following options:

du <options> <directory or file>

<options>

-k This command shows a summary of the space available for each mounted file system.

-h This command shows the output in a human readable format (*kilobytes, megabytes, gigabytes*)

<directory or file>

<no argument> If no argument is given, all file systems will be shown.

/MountPoint This shows the amount of free and used space in a file system. The mount point is specified to indicate which file system is being checked. If no mount point is given, information is given on all mounted file systems.

/dev/dsk/c0t0d0s0 The logical device name. In this case, the root partition of the first disk (**c0t0d0s0**) is used as a reference example.

Lesson 15.3 Using the **df** Command

The **df** command shows how much space is used and available on a file system. This lesson examines the various outputs that can be achieved with the **df** command. The **ff** command is also shown in this lesson as a quick reference.

1. Log in as the root user.
2. Open a Terminal window
3. Type the command **df**
This shows a summary of all the free and used blocks on a hard drive.
4. Type the command **df -k**
*This shows a summary of all the free and used blocks on a hard drive in kilobytes. The format of **df -k** is different than the **df** command without arguments..*
5. Type the command **df -h**
This shows a summary of all the free and used blocks on a hard drive in human readable format This is a new feature in Solaris 9. The human readable format uses words like kilobytes, megabytes, gigabytes instead of the difficult blocks format.
6. Type the command **df -k /**
This command shows only the space used on the root (/) slice
7. Type the command **df -k /dev/dsk/c0t0d0s0**
*This demonstrates that the **df** command can reference a slice by its logical device name.
/dev/dsk/c#t#d##s#.*
8. Type the command **ff /dev/dsk/c0t2d0s0**
*The **ff** command produces a list of pathnames and inode numbers of files in the file system. This command take a very long time to work.*

The quot Command

The **quot** command is used to show how much space each user is taking on the file system. Without any options, this command shows how much disk space (in kilobytes) is being used by each user
The **quot** command supports the following options:

quot <options> <directory or file>

<options>

quot -af Shows disk space used by users on all mounted file systems

quot -f /dev/dsk/c0t0d0s0
 Shows the number of files and space used on a specific file system

Lesson 15.4 Using the quot Command

In this lesson, the **quot** command will be used on a very small file system. It shows how much space is being used by every user. This is critical if a file system becomes full and needs to be cleaned out.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command **quot -f /dev/dsk/c0t0d0s0**
This command shows the number of blocks each user is taking on the c0t0d0s0 file system, the number of files and the name of the user.
4. Type the command **quot -af**
The command **quot -af** shows the space used on all mounted file systems.

The **fsck** Command

This is a very powerful tool for repairing bad disks. It checks the consistency of a file system. If a file system uses UFS logging, it will be more resistant to file system corruption. The **fsck** command checks the following items in a file system.

- Lost files and directories Puts files and directories that are unallocated in the **lost+found** directory. An inode number is assigned to these fragments..
- Superblock Checks the Superblock for inconsistencies
- Inode inconsistency Checks whether inodes reference the correct data blocks, names, etc.
- Data block consistency Checks directory data blocks. Inodes pointing to unallocated blocks are tagged as used.
- Cylinder group block consistency Checks unallocated data blocks claimed by inodes, the unallocated data block count ,and the inode count.

The **fsck** command goes through five phases of checking:

1. Check blocks and sizes – checks inodes for consistency
2. Check pathnames – checks directory inode consistency
3. Check connectivity – checks that all directories are attached to a file system
4. Check reference counts – compares link count information from 2) and 3) for errors
5. Check cylinder groups – data problems related to cylinder group’s information.

The **fsck** command is run in two different modes.

Non-interactive mode Preen or silent mode, repairs only minor inconsistency problems. When the Solaris Operating system is started it checks all the file systems in the **/etc/vfstab** file that are marked for an **fsck** scan (any file with a number greater than 0 in the FCK CHECK column.). If there are no severe problems, the **fsck** command remains in preen mode.

Interactive mode In this mode the **fsck** command asks for **yes** or **no** responses for its suggestion on how to correct the problem. If a sever file system problem occurs, **fsck** leaves preen mode and runs in interactive mode. This is where a system administrator has to decide how to repair a file system.

The **fsck** command can be run with various options such as:

fsck /dev/rdisk/c0t0d0s7 The best way to check a file system is to check the file system’s logical device file, and not in the **/etc/vfstab** file.

fsck /opt Checks the file system mounted by the **/opt** directory only if its mount point is defined in the **/etc/vfstab** file.

fsck -f Force a file system check.

fsck -p Preen mode, fixes file system non-interactively (preen) required to enable parallel file system checking.

fsck -o b=32 /dev/rdisk/c1t3d0s0
 The **-o b=** option is used to replace a bad superblock with a superblock backup. The command **newfs -N** will show the location of backup superblocks used with this command.

Lesson 15.5 Using the fsck Command

The **fsck** command is used to check the condition of a file system. In this lesson the root (/) file system will be checked with the **fsck** command.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command **fsck /dev/rdisk/c0t0d0s0**
Type Y to answer the warning message about having a mounted file system

The newfs Command

As mentioned earlier, the **newfs** command is used to create a new file system. This command supports the following options:

newfs -N /dev/rdisk/c#t#d#s# Used to shows all the backup superblocks

Lesson 15.6 Using the newfs Command

The **fsck** command is used to check the condition of a file system. In this lesson the first slice of the second disk is used to demonstrate the **newfs** command. Understand that when the **newfs** command is run, it will create a new file system on a slice and destroy the previous file system. All files on the previous system will be wiped away. The **newfs** command was covered in Chapter 14.

WARNING

This lesson uses the **newfs** command. This command will destroy any data on the slice on which it is creating a new file system. If you are unfamiliar with the Solaris disk device naming conventions (example: **/dev/dsk/c0t3d0s4**) re-read the "Solaris Disk Naming Conventions" section in Chapter 13.

When the **newfs** command is used, it will destroy the file system on the second hard drive. Copy any critical information from the second hard drive to another source before running the **newfs** command.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command **newfs /dev/rdisk/c0t2d0s0**
Type Y to the "are you sure" message.
4. Type the command **newfs -N /dev/rdisk/c0t2d0s0**

*The command **newfs -N** shows information about a file system, such as the number of sectors, cylinders and tracks. It also shows the locations of the backup superblocks.*

Key Points to Remember

This chapter was designed to show the various file systems available on Solaris 9. The UFS and NFS file systems are the most common ones found on Solaris servers. The most important utility that a system administrator must know how to use is the **fsck** command. This command will repair a damaged file system (depending on the amount of damage.) It is very important that the system administrator understand the mechanics of a file system before using the **fsck** command. For example, the **fsck** command can restore the Superblock from an alternate cylinder group. A system administrator needs to know what is a Superblock and what is an alternate cylinder group to effectively use the **fsck** command.

Chapter 16 TCP/IP on Solaris

Lessons in This Chapter

Lesson 16.1 Using the <code>ifconfig</code> Command.....	16-11
Lesson 16.2 Change the Network Settings.....	16-11
Lesson 16.3 Using the <code>banner</code> Command.....	16-13
Lesson 16.4 Using the <code>ping</code> Command.....	16-13
Lesson 16.5 Using the <code>snoop</code> command.....	16-16

Introduction

A computer network operates just like the postal system. But instead of delivering physical packages, a computer network delivers packets of electronic information. Just as every house, factory and office building has an address, each network device (workstation, server, router, ftp server) has an electronic address. Instead of a postal delivery address (1235 Johnson Street) each device on a network has a network address that looks something like this (203.14.207.14). There can be different types of networks that can communicate with each other, just as the U.S. postal service can send and receive mail from the Russian postal service.

This chapter describes how to set up Solaris 9 on an IPv4 (IP version 4) network. This does not cover IPv6 issues. Chapter 27 describes IPv6 networking issues.

Common Terms

Before we describe networking on Solaris 9, the reader should be familiar with the following networking terms. Computer networks are generally classified by the size and scope of the network.

Types of Networks:

LAN (Local Area Network)

This is a network that resides within a building or a floor of a building. It can contain a single network or a collection of networks.

WAN (Wide Area Network)

This network covers a city or is between cities or nations. Satellite receivers, routers, microwave dishes, and communications hubs are the typical kinds of equipment found within a WAN architecture.

Network Media

This describes the cables that connect the network devices to each other. Network media can also include non-cable media, including microwave channels, radio channels and infrared devices.

Network Cables:

Telephone wires

Used for dialup connections with modems. They have a very slow data transfer rate.

- Thin coaxial cable** Looks like the cable used with cable TV. Almost never used in networks anymore. 100 Mbps transfer rate.
- Thick coaxial cable** Also looks like the cable used with cable TV. Primarily used as the backbone of a network. 100 Mbps transfer rate.
- Twisted pair cable** Commonly used with networks. It looks like telephone wire but it is thicker. It has a connector that looks like a telephone jack connector except that it has an 8-pin plug and it is wider than the standard telephone jack connector.
- Fiber optic cable** Uses glass strands instead of wires for communications. Good for environments that have a lot of “magnetic noise.” Also, fiber optic cables can transfer information over a long distance without the need for a repeater.
- Packet** The unit of information that travels over the network media. The packet contains a header, data and trailer. The header usually contains control information (what computer to go to, TCP port number, MAC number, etc). The Data section has the information to be sent. The Trailer usually has error correction codes (ECC) that check to make sure the packet was not corrupted along the trip. Figure 16.1 shows an Ethernet packet.

Packet Header (TCP Port #, Mac #, Destination address)	Data (such as a GIF image)	Trailer (CRC error correction check)
--	----------------------------	--------------------------------------

Figure 16.1 An Ethernet Packet

- Network Protocols** People use languages to communicate with each other. For example, we have English, French and German as languages. Computers also use “languages” to communicate with each other. (TCP/IP, IPX, and NetBEUI are examples of network protocols.)
- Network Layers** Network protocols are broken down into layers. Each layer in a protocol “talks” to a layer above or below it. The reason that network communications is broken down into layers is so that different manufacturers can create different software and hardware that can communicate with the other layers of the network model. This is useful so that if a company creates network cards, its card will work with different types of cables and operating systems.

The most common protocol used within networks is the TCP/IP protocol (Transmission Control Protocol/ Internet Protocol). The TCP/IP protocol covers two main concepts. The TCP part of the protocol covers communications within a computer. The IP part of TCP/IP deals with the routing of information between computer networks.

There are two theoretical models that are used to describe the TCP/IP protocol. These are covered below.

The OSI Model and TCP/IP Model

One common question students ask instructors about the OSI model and networking models is “Why should I study the OSI model? This stuff is really boring, and I will never use the material” The answer is simple. A competent system administrator should understand both the OSI and TCP/IP models, so that he or she can work with networking equipment. Most instructors will ask the student a questions like “If a router arrives and it supports Transport layer functions, what does that mean? Can you install the router with IPv4 and IPv6? If another server can not respond to ping requests, but does accept FTP connections, is this an Ethernet problem or a TCP problem?” In a nutshell, because a system administrator will work with networking equipment and network engineers, he or she must be able to understand networking terms and networking models.

There are two predominant networking models. These are::

OSI Model The first model was created by the International Standards Organization (ISO). It is called the OSI (Open Systems Interconnection) model.

TCP/IP Model The other model is named the TCP/IP model. This is also a theoretical model that describes the TCP/IP protocol.

Quick Tip

- The biggest problem that readers tend to face is the fact that both models use the same terms. For example, in both the OSI model and the TCP/IP model the term “Transport Layer” is used, even though the terms deal with different concepts.
- Another problem comes from the fact that both models describe TCP/IP networking, and one model is named The "TCP/IP model." Both models describe the exact same thing, so do not get hung up about the model names.

The OSI model

The Open Systems Interconnection (OSI) model was created by the International Standards Organization (ISO). There are seven layers to this model. When new hardware is purchased, such as a 3com™ router or Nortel Networks™ bridge, the hardware documentation references this model. The documentation might say something like “This router supports packets sniffing at the Transport layer.”

This model basically describes how TCP/IP works. The highest level of the model is the Application layer, while the lowest layer of the model is the Physical layer.

Application Layer This layer only deals with the end user’s software. If a company wants to create an FTP application, it should try to conform to the OSI model’s standards for FTP. Imagine if an FTP GET command on one computer did something different than the FTP GET command on another computer. Commonly used Application tools are telnet, ftp, email, and HTTP. The Application layer describes the applications that users have at their disposal.

Presentation Layer This layer makes sure the data sent and received to the Application layer software is the same. For example, if two computers exchange email, they both need to use the same text characters (ASCII, Chinese Characters, Korean letters.) The Presentation layer ensures that both pieces of software are using the same data formats. Imagine if one computer sent Chinese characters and another computer tried to print these characters in English. The Presentation layer does not only cover text characters. It can also cover other types of

data formats (encryption methods, word lengths, database fields). Its only job is to ensure that two applications are “talking” in the same language.

The next five layers of the OSI model deal with the sending and receiving of electronic packages. The easiest way to describe these layers of the OSI model is to use an analogy of two warehouse managers trying to send packages to each other through the postal service.

Session Layer The Session layer acts just like the two warehouse managers talking to each other. The communication would sound something like “Only send 1400 packages a day...my warehouse is full, so hold off on the next shipment... I can receive some more packages today...” The session layer within the OSI model performs communication flow control. If one of the two computers is being overrun with packets, the Session layer will contact the other computer and tell it to slow down its transmissions. The Session layer performs the same negotiations, except that it deals with electronic packages, not postal packages.

Transport Layer Using the same shipping analogy, the transport layer acts like an internal shipping clerk within the warehouse. Mail comes into a central mail/receiving area in a company. The shipping clerk sends different pieces of mail to different people within the corporate site. Some mail would go to Building 3 Secretary 4, some mail would go to Building 3 Secretary 7, and some mail would go to building 6 Management Office 3. The Transport layer does the same thing. It sends information to the proper application. Imagine if a user had three telnet sessions open between his workstation and a server. The Transport layer would make sure the data packets from the three telnet sessions would go to the proper telnet session. The TCP and UDP protocols reside at this layer.

So now the package leaves the warehouse and goes into the postal mail system...

Network Layer Imagine if you had a rather poor-quality postal service that could deliver mail to a specific street but could not deliver mail to a specific warehouse. The Network layer works just like the poor-quality postal service. If a message were sent to 234 Anderson Lane, the Network layer would only deliver it to the street named Anderson Lane. Unfortunately, the postal worker on Anderson Lane would not know any of the houses. He would walk down the road holding the package in his hand. The occupants of the homes would then look at the label on the package. They would then pick up and read any package addressed to them.

The Network layer does not know what devices are on a network segment. It only sends the Ethernet packet down the wire. Each computer looks at the packet to determine if the packet is for it. The network layer uses electronic addresses called an IP address. This looks something like 203.24.13.86. This layer is only responsible for routing packets between network segments. The actual pickup of packets on the network segment is done by the client computers.

Let’s try a simple example. Figure 16.2 is an illustration of a simple network. A router is connected to the Internet.

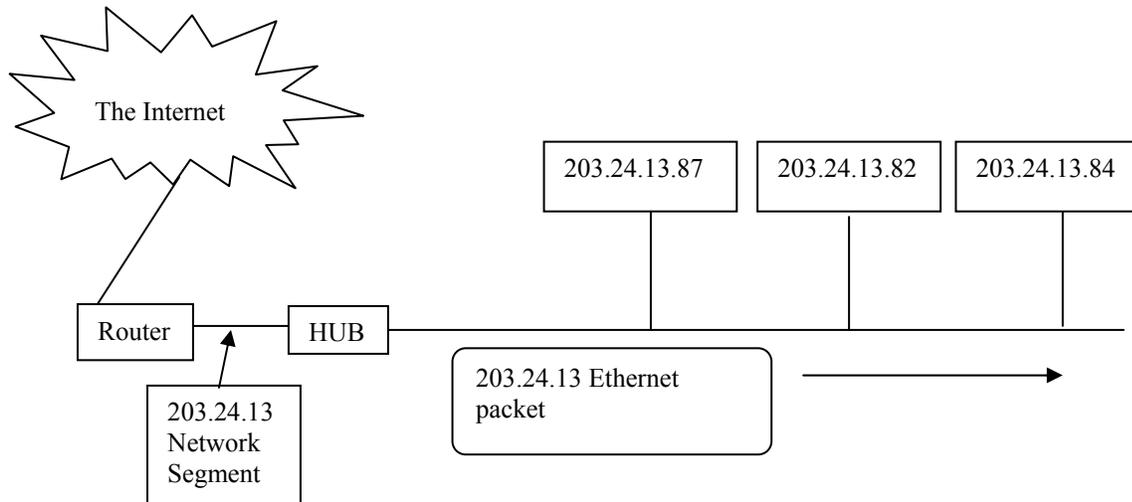


Figure 16.2 A Simple Network

1. On the inward side is the network segment 203.24.13. Somewhere out in the Internet, a computer is trying to reach the server with the IP address of 203.24.13.84 on the network.
2. The IP address of 203.24.13.84 comes through your router. The router works at the Network Layer. It passes any traffic through with the IP address of 203.24.13. But it does not know anything about the .84 last digit.
3. All the computers on the LAN see the 203.24.13.84 packet. But only the server with the IP address 203.24.13.84 will process the packet. The other servers (with the IP addresses of 203.24.13.82 and 203.24.13.87) will also see the packet. But they will ignore it, because the IP address does not match their IP addresses. The golden rule is: routers pass traffic onto a network segment only, while each computer on the network segment picks up data sent to its specific IP address.

Data Link layer

This layer is used to sort through the street mail and pick only the packages intended for 234 Anderson Lane. The Data Link Layer does not know what "Anderson Lane" means. It only understands "234." It would pick up all the packages for warehouse 234 and ignore packages for warehouse 235 and 273. The Data Link layer only understands the Ethernet address of its host computer. It does not know anything about IP addresses, or FTP, or any other TCP/IP issues. It only reads Ethernet addresses and picks up data packets designated for its host.

Physical Layer

The Physical layer only deals with physical delivery of the mail. It would have specifications such as: "The postal truck can only be 8 feet wide, and the driver can not drive faster than 30 miles per hour." The Physical layer only deals with physical specifications such as voltages, and bits per second. When a company makes a network device, such as an Ethernet card, it must make sure that its hardware uses the same voltages, signal strength and timing as other network devices. It must also check its design specifications against the OSI model to make sure that its equipment conforms with all other network devices. Imagine if a company made an Ethernet card that used a signal strength of 200 volts and 30 amps. It would fry every other network device on the LAN!

Let's look at some common network equipment that is used to create a network, and at how these pieces of equipment correspond to the OSI layers.

Network Interface Card (NIC)

Every device that is connected to a network must have a Network Interface Card. A NIC is very similar to a modem, except that it uses Ethernet cables instead of telephone wires. The NIC card can be designed to communicate over any network media (thin coaxial cable, thick coaxial cable, fiber optic cable). The card can also be built into a motherboard, or it can be an add-on expansion card.

Some terms associated with a NIC include:

MAC Address

A MAC address is similar a Social Security ID #. Every NIC card produced has a unique MAC address. This is a 12-digit hexadecimal number that represents the NIC card. A MAC address looks like:

2a:3b:12:21:4d:3b

Ethernet Port

This is a female port connector on the NIC, that looks like a telephone jack. Most of the time it is an RJ-45 connector, but it can be any connector the card designer can produce (such as a fiber optic connector or vampire clamp).

Repeater

This can only be described as a dumb signal regenerator. A repeater only has two ports. It takes a signal in from one cable, strengthens the signal, cleans the signal up, and then sends the signal out the other port. A repeater does not know anything about TCP/IP, MAC Addresses, or any other network concepts. It is primarily used in electronically noisy environments, or when a signal needs to be boosted to travel a longer distance. Figure 16.3 is an illustration of a repeater cleaning a weak signal. Repeaters are starting to become hard to find. Most network engineers are purchasing a hub or switch, instead of a repeater.

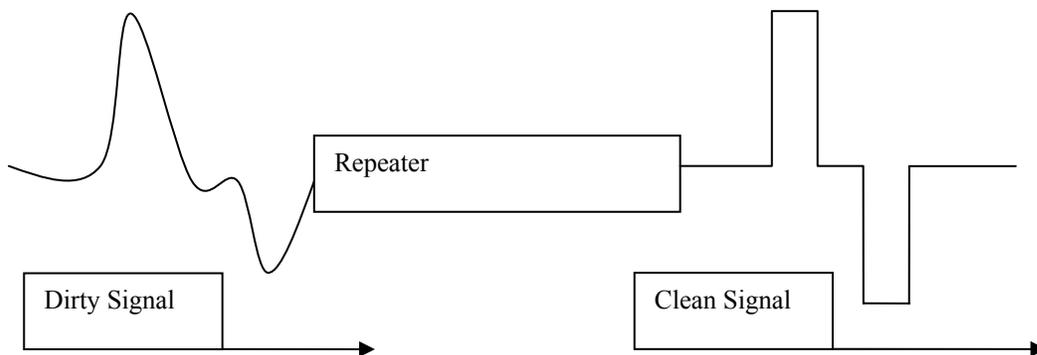


Figure 16.3 Repeater Cleaning a Signal

HUB

A hub is a repeater with multiple ports. It takes any signal it receives on any port and broadcasts that signal out on all ports. A hub is useful in peer to peer networks where all the computers on the network need to “talk” with each other. Just like a repeater, a hub does not understand networking terms. It is just a dumb signal regenerator that broadcasts all traffic to all the ports.

Switch

A switch works just like a hub, except that it analyzes the network traffic on each port and determines “who is try to talk to who.” Unlike a hub that broadcasts all traffic it receives from any port to all other ports, a switch only transmits packets to the port that needs to receive the information. A switch works at the Data Layer of the OSI model. It looks at MAC addresses and only routes network traffic between two computers that are talking to each other.

Router Up to this point we have described network devices that all work on a local network. Obviously, the sun.com network is not connected directly to the uswest.com network. For independent networks to talk with each other, routers are needed. Routers work on the Network layer of the OSI model. They send IP packets from one network (such as the sun.com network) to another (such as the uswest.com network)

Gateway A gateway is an even more sophisticated device that is used to let one type of network talk to another type of network. For example, a TCP/IP network can talk to an IPX network, and to a Macintosh Apple Talk network, if it has a gateway that understands all three protocols (TCP/IP, IPX, and Apple Talk). The gateway operates on the Session Layer of the OSI model.

Application Layer Programs

The OSI model application layer deals with the applications that the end user and other programs use to communicate over a network. **FTP** is an example of an Application layer program that users can run, and that can be called on by programs in the background. **Telnet** is an application that is generally used only by a user.

The point is that some Application layer programs can be called by a user and some can be called by both users and programs. Some common Application layer programs are:

telnet	Used to send text messages between two computers; commonly used to log on to one computer from another
FTP	Used to transfer files between two computers
rlogin	Allows a user to remotely log in to another computer
rcp	Used to copy files remotely between two computers; much like FTP
rsh	Starts a remote shell on another computer
rexec	Used to execute a command on a remote computer
rwho	Shows which users are active on a foreign computer
finger	Shows which users are currently logged on a local or remote computer
rup	Shows the status of a remote computer
ping	Sends an echo request to a remote system and receives an echo reply
spray	Sends a whole bunch of network packets at another computer; used to test network connections

Other Network Concepts

Name Service

It would be rather fatiguing to try and remember IP addresses (the main server is 203.13.47.307, the email server is 203.13.47.309, etc). To make it easier for humans, the **/etc/hosts** file lists both the human readable address and IP address for various hosts. Figure 16.4 is a typical **/etc/hosts** file.

100.100.100.100	sun100 loghost
103.100.102.103	data32
232.43.83.109	workstation3
203.13.47.309	dog

Figure 16.4 An /etc/hosts file

This file allows users to use an easy-to-remember *hostname* instead of an IP address. For example, given the last line in Figure 16,4, a user could use the command **ping dog** to test the connection to the network computer at IP address 203.13.47.309.

Unfortunately, with very large networks that have thousands of computers, all the computer names and their corresponding IP addresses can not be kept in a local text file. The solution comes in the form of *Name Servers*. These are network servers that hold all the hostnames and IP addresses in a network domain (such as a company). If a user wanted to use the command **ping dog** and the local **/etc/hosts** file did not have an entry for the hostname **dog**, the operating system could get this information from a name server. Using the function called *Name Service*, the name server sends back the IP address that corresponds to **dog**.

One very common type of name server is the "Domain Name Service server." This is usually shortened to "DNS Server." Solaris 9 also supports NIS, NIS+ and LDAP for name service resolution.

IP addresses An IP address is a series of numbers that represents any network capable device, for example 203.13.24.209 is an IP address. A company can not connect its network to the Internet and simply assign itself any IP address desired. All IP addresses and domain names must be registered with an organization named InterNIC. InterNIC has several categories of domain registration. A registration is dependent on the organization's size and need.

There are 5 major categories of IP addresses.

Class A address This is used for extremely large networks. There can be millions of hosts. The first 8 bits represent the network ID. Only 1-126 are usable IP addresses. The number 127 is reserved, and IP addresses above 127 are also reserved. The remaining 24 bits represent the host ID's. IP addresses of 0 and 256 are reserved, so a total of 16,777,214 hosts can be used.

Class B address This is used for medium-sized networks and companies. The first 16 bits are used to identify the network, and the last 16 bits are used to identify the host. The numbers 128-192 are valid IP addresses. A class B network can accommodate 65,534 hosts. The number 0 is reserved for a network number, and 255 is reserved a broadcast number.

Class C address These are used for very small companies. Class C networks can only accommodate 254 hosts. The first 24 bits represent the network, and the last 8 bits represent the hosts. The numbers from 192-223 are the first IP addresses of a class C network.

Class D address These are used by routers and WAN equipment to talk with each other. The first IP addresses are in the range 224-239.

Class E address These are used for experimental purposes and cover the first IP address range 240 – 255. Ordinary companies can not access or use these IP addresses.

The TCP/IP Model

In this chapter, we have discussed the OSI model at length. The other model that is used to describe TCP/IP is the "TCP/IP Model," which is discussed below.

The TCP/IP model has five layers:

- **Application Layer**
- **Transport Layer**
- **Internet Layer**
- **Network Interface Layer**
- **Hardware Layer**

Basically the TCP/IP model is very similar to the OSI model. One of the biggest things that confuses readers is that both models use the same term "Transport Layer." The Transport Layer in the TCP/IP model covers the Presentation, Session and Transport layer of the OSI model. The other layers of the model correspond to the OSI model's layers. Figure 16.4 shows how the TCP and the OSI models relate to each other.

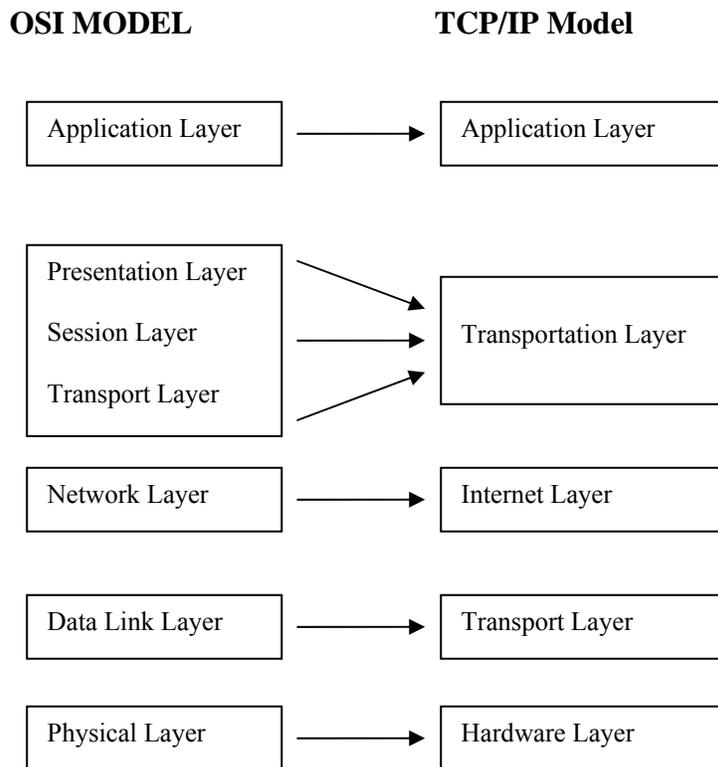


Figure 16.4 OSI and TCP/IP Models

Networking Files on Solaris 9

There are several text files that keep the network configuration information on Sun Solaris 9:

/etc/hosts This file saves the hostname to IP addresses table information.

/etc/hostname.<interface>

This file sets a hostname to an interface. For example, say that a system has two network cards on a server. The device aliases for these cards are **hme0** and **hme1**. A server would then have two files inside the **/etc** directory: **/etc/hostname.hme0** and **/etc/hostname.hme1**. These files contain the hostname that applies to each of the NIC cards.

/etc/nodename This text file contains the hostname of the computer.

/etc/defaultdomain

This text file contains the name of the domain that the computer resides on, for example: **uswest.com**

/etc/defaultrouter

This text file contains the IP address of the default router. If the server determines that a host is not on the local network, this file will indicate where to send the IP packet. It is the router's job to route IP packets that are outside the local network.

Understanding the ifconfig Command

The **ifconfig** command is used to view and set up an Ethernet card's configuration. There are two versions of **ifconfig**. When Solaris 9 boots, it uses **/sbin/ifconfig** to set up the Ethernet card. The root user can also use the **/usr/sbin/ifconfig** command to reconfigure the Ethernet card. Ordinary users are not allowed to use **ifconfig**.

ifconfig -a This shows all the Ethernet cards and their information. For example:

```
eri0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPV4> mtu 1500 index 2  
inet 192.168.0.4 netmask fffff00 broadcast 192.168.0.255  
ether 0:3:ba:4c:c1:3d
```

eri0 The name of the Ethernet card. **eri** – the device name, **0** – first instance of the Ethernet card. If a server had four Ethernet cards, the **ifconfig** command would display: **eri0**, **eri1**, **eri2**, and **eri3**.

flags These numerically represent the same things as the **<UP,BROADCAST,RUNNING,MULTICAST,IPV4>** section. Don't worry about these numbers; the operating system uses them.

UP This Ethernet card is up (Maintenance mode can be up or down)

BROADCASTING

Card is broadcasting its presence on the network, using broadcast packets.

RUNNING

The Ethernet card is running.

MULTICAST

This Ethernet card accepts multicast addresses. A multicast address sends Ethernet packets to multiple computers at the same time, hence the name multi (several) cast (send). This is a special Class D address.

IPv4 Using IPv4. This is the standard "xxx.xxx.xxx.xxx" type of address (example 126.23.203.14).

mtu 1500

Maximum Transmission Unit, the size of the Ethernet packet that this system will take.

index Shows how many Ethernet devices are using this setting.

inet The IP address of the Ethernet card.

netmask

This shows the netmask in hexadecimal numbers. Since f (hex) = 15 decimal, the value **ff** = 255 [(15*16) + (15 * 1)].

broadcast

The broadcast address this system uses. When the Solaris operating system needs to contact other computers on the same network segment it uses a broadcast address.

ether The Ethernet address of the NIC card. The Ethernet address is different for every card in the world. There are no two cards that are exactly the same.

Lesson 16.1 Using the **ifconfig** Command

In this lesson the readers will use the **ifconfig** command to view the different Ethernet cards and configurations on their workstations.

1. Log in as the root user.
2. Open a Terminal window.
To open a Terminal window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Tools menu item, then left click on the Terminal icon.
3. Type the command **ifconfig -a**
*The **ifconfig** command should display an output like this:*

```
# ifconfig -a
```

```
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
    inet 127.0.0.1 netmask ffffffff
eri0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
    inet 192.168.0.4 netmask fffffff0 broadcast 192.168.0.255
    ether 0:3:ba:4e:c2:2a
```

lo0 (little L, The letter 0 and zero) is a test loopback Ethernet interface; it does not exist.

eri0 is the device name for the Ethernet card on a SunBlade 100.

4. Type the command **ifconfig eri0 down**
This disables the Ethernet card. This command does not ordinarily display any output.
5. Type the command **ifconfig eri0 up**
This enables the Ethernet card again. This command does not ordinarily display any output.

Lesson 16.2 Change the Network Settings

In this lesson the reader will use the **ifconfig** command to temporarily change the IP address of the system. These changes are not permanent and will only remain until the workstation is rebooted. To make the new IP address permanent, files in the **/etc** directory must be changed.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command **ifconfig -a**
*This shows all the networking devices on the system. Pick the name of the Ethernet card to change the IP address. The name **eri0** will be used for this example.*
4. Type the command **ifconfig eri0 down**
This temporarily disables the Ethernet card.
5. Type the command **ifconfig eri0 192.168.0.9**
This changes the address of the Ethernet card. Change the Ethernet card's IP address to an unused IP address on the network segment.
6. Type the command **ifconfig eri0 netmask 255.255.255.0**
This changes the subnet mask on the Ethernet card.
7. Type the command **ifconfig eri0 up**
The brings the network card back online.
8. Type the command **route add net default 192.168.0.2**
This command changes the default router's IP address. The output should look like:
add net default: gateway 192.168.0.2
9. To gracefully reboot the system, type the command **init 6**
The original network settings should not be affected by this lesson, if for some reason the new settings have taken hold, follow the lesson again but change the network settings to good settings.

OPTIONAL

To make these changes permanent, edit these files:

/etc/hosts

sun100 192.168.0.4 change to **sun100 192.168.0.9**

/etc/netmasks

192.168.0.0 255.255.255.0 change to **192.168.0.0 255.255.255.128**

/etc/defaultrouter

192.168.0.1 change to **192.168.0.2**

The banner Command

The **banner** command was also discussed in detail in Chapter 1. This is an OpenBoot command that is typed at the **OK** prompt. The **banner** command can only be used when the system is at run level zero (**0**). Figure 16.5 shows a sample display from this command:

```

Sun Blade 100 (UltraSPARC-IIe), Keyboard Present
OpenBoot 4.0, 128 MB memory installed, Serial # 50643816
Ethernet address 0:3:ba:4b:c4:5d, Host ID: 8034d13b

```

Figure 16.5 Output from the Banner Command

The **banner** command displays the following output:

Model	The system's model name, including type of CPU
Keyboard	Whether a keyboard is present
OpenBoot Version	The system OpenBoot version

Memory	The amount of physical RAM detected.
Serial #	The system serial number.
Ethernet Address	The Ethernet address on the NVRAM chip.
HostID	A special number used for software licenses

Lesson 16.3 Using the **banner** Command

This is a very simple demonstration lesson that shows how to use the **banner** command at the **OK** prompt.

*Bring the system down to run level zero (0).
If the operating system is not running, skip step 1 through 3.*

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command **init 0**
If the operating system is not running, start the SPARC-based computer and immediately press the STOP + A keys simultaneously.
4. At the **OK** prompt, type the command **banner**

The **ping** Command

The **ping** command is used to test network connections. This command only tells a system administrator if two servers can communicate over the network. The **ping** command can use the foreign host's IP address or its hostname (provided that the local system knows the hostname.) For example:

Assume that you want to test the connection to a system with the IP address **204.35.18.23**, and that the hostname for this server is **dataserver23**. The following examples show the two ways you could ping that system, and what the response will be if the command succeeds.

```
ping 204.35.18.23
204.35.18.23 is alive
or
ping dataserver23
dataserver23 is alive
```

Lesson 16.4 Using the **ping** Command

In this lesson the readers will **ping** other computers, by their IP address and hostname. They will then disconnect their computer from the network and try the **ping** command again.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command **cat /etc/hosts**
*Write down the IP address for your workstation.
Find the IP address of another workstation, write it down.*
4. Type the command **ping 192.168.0.1** or
ping <IP_address_of_this_workstation>
*This gives a rather dry response, such as **192.168.0.1 is alive**.*
5. Type the command **ping -s <IP_address_of_another_workstation>**
This gives a much better response, showing the packets sent and the time the packets arrived back.

The output should look something like this:

```
# ping -s sun100
PING sun100: 56 data bytes
64 bytes from sun100 (192.168.0.4): icmp_seq=0. time=0. ms
64 bytes from sun100 (192.168.0.4): icmp_seq=1. time=0. ms
64 bytes from sun100 (192.168.0.4): icmp_seq=2. time=0. ms
64 bytes from sun100 (192.168.0.4): icmp_seq=3. time=0. ms
^C
----sun100 PING Statistics----
4 packets transmitted, 4 packets received, 0% packet loss
round-trip (ms) min/avg/max = 0/0/0
```

6. Press CTL + C to end the `ping -s` test.

TCP Ports

Different applications use different TCP ports. For example, `telnet` uses TCP port 23 and `whois` uses TCP port 43. There are times that a network engineer will ask the system administrator to set up a program to use a specific port, such as 3859. The `/etc/services` file is used to show what port is being used by what type of communications.

Figure 16.6 shows a typical `/etc/services` file found on a Solaris workstation.

```
tcpmux      1/tcp
echo        7/tcp
echo        7/udp
discard     9/tcp      sink null
discard     9/udp      sink null
systat      11/tcp     users
daytime     13/tcp
daytime     13/udp
netstat     15/tcp
chargen     19/tcp     ttytst source
chargen     19/udp     ttytst source
ftp-data    20/tcp
ftp         21/tcp
ssh         22/tcp     # Secure Shell
```

Figure 16.6 The `/etc/services` file

The `inetd` daemon

There are certain types of network traffic assigned to each network port. For example, `telnet` sessions come in through port number 23. `FTP` connections come in through ports 20 and 21. However, it would be very inefficient for a server to have the `telnet` and `ftp` daemons running all the time, even when there is no network that needs their services.

Solaris 9 uses the `inetd` daemon to handle TCP port issues. This daemon listens for network traffic on each port. When a request comes in for an FTP service, the `inetd` daemon wakes up the `ftp` daemon and the FTP session starts.

The `inetd` daemon is configured through the `/etc/inet/inetd.conf` file. Figure 16.7 shows a sample of a typical `/etc/inet/inetd.conf` file.

```
# TELNETD - telnet server daemon
```

```
telnet stream tcp6 nowait root /usr/sbin/in.telnetd in.telnetd
# FTPD - FTP server daemon
ftp stream tcp6 nowait root /usr/sbin/in.ftpd in.ftpd -a
```

Figure 16.7 The /etc/inet/inetd.conf File

This file is created when Solaris 9 is initially installed. Some 3rd party software can change this file as well as the root user.

As can be seen in Figure 16.7, the file only gives the name of the service and what program is to be started if a request comes in for that service. If any changes are made to this file or to the /etc/services file, the command:

```
pkill -HUP inetd
```

must be run to restart the **inetd** daemon.

The **inetd** daemon will then re-read the /etc/inet/inetd.conf file.

RPC (Remote Procedure Calls)

One of the problems that Sun encountered with the **inetd** daemon is that a lot of software developers and software companies all wanted a particular port reserved for their application. This became unmanageable! A solution was found with the RPC service. The RPC services is a client/server service that runs under the port 111.

Here is a sample RPC session (demonstrated as a conversation).

Client A's RPC daemon says: *"I have an application, Ben's Client Database Tool, that needs to talk with the Ben's Server Database Software, but I don't know what port to use."*

Server A's RPC daemon says: *"Ok, I will contact the **rpcbind** process in this server."*

The server's RPC process says: *"I will assign port 100032 for Client A to talk with Ben's Database Software. I will also write in the /etc/rpcinfo file that Client A is talking to the Server Database Software on port 100032"*

The server's RPC daemon says: *"Hello, RPC daemon on Client A, you can use port 100032 to talk with the Server Database Software"*

Client A's RPC daemon says: *"OK, I will tell the Client Database Tool to use port 100032 for the database communications"*

Basically, the RPC daemon on the client and the RPC daemon on the server open up a port for communications. This eliminates the need to assign a specific port for a specific application.

To check for a specific port number, type the command:

```
rpcinfo -p <client host name>
```

If for some reason an RPC service has problems, it can be stopped and restarted to establish a new connection. The following series of commands is used to perform the stop/restart functions:

```
rpcinfo -d <service name>
rpcinfo -p | grep <service name>
pkill -HUP inetd
```

```
rpcinfo -p <service name>
```

The snoop Command

This command is used to pick up network traffic and record the traffic for further analysis. The **snoop** command supports the following arguments:

```
snoop <options> <IPaddress or hostname>
```

<options>

- a** Produces audio clicks for each packet that arrives.
- i <filename>** Recovers the output from the file.
- o <filename>** Sends the output to a file.
- v** Produces detailed verbose output.
- V** Produces less detailed verbose output.

Lesson 16.5 Using the snoop command

This lesson shows how the **snoop** command picks up and reports on network traffic.

1. Log in as the root user.
2. Open a Terminal window.
3. On your computer, type the command **snoop <IP address of foreign computer>**
*If the foreign computer has the IP address of 100.100.100.105 type the command **snoop 100.100.100.105**.*
4. On the foreign computer, type the command **ping <your IP address>**
*When the foreign computer pings your computer, the **snoop** command should start to show network traffic on your screen.*
5. On your computer, press CTL + C.
6. On your computer, type the command **snoop -a <IP address of foreign computer>**
*If the foreign computer has the IP address of 100.100.100.105 type the command **snoop -a 100.100.100.105**.*
7. On the foreign computer, type the command **ping <your IP address>**
There should now be audible clicks that sound like a Geiger counter from your Solaris computer.
7. On your computer, press the CTL + C keys simultaneously.
*This will end the **snoop** session.*
8. On your computer, type the command **snoop -o mycapturefile <IP address of foreign computer>**
*If the foreign computer has the IP address of 100.100.100.105 type the command **snoop -o mycapturefile 100.100.100.105**.*
9. On the foreign computer, type the command **ping < your IP address>**
10. On your computer, press CTL + C.
11. On your computer, type the command **snoop -i mycapturefile | more**
*This shows the output of the **snoop** command.*
12. (From here on, all commands are on your computer.) Type the command
snoop -i mycapturefile > /textcapturefile
*This copies the output of the **snoop** command to a text file that text editors can use.*

8. Type the command **vi /textcapturefile**
Look through the file for information on the TCP/IP traffic from your neighbor's computer to your computer.
13. Type the command **rm mycapturefile**
14. Type the command **rm textcapturefile**
The last two commands cleaned the lesson files off of your workstation.

Key Points to Remember

This chapter introduced the reader to IPv4 networking concepts. This chapter went into great detail about the OSI and the TCP/IP model. It is very important that a system administrator know the different layers of these models to work in industry. This chapter also described how to work with Ethernet cards on SPARC systems. The key command to understand from this chapter is the **fsck** command. It can be used to show an Ethernet card's settings or change an Ethernet card's settings.

Chapter 17 Printing in Solaris

Lessons in This Chapter

Lesson 17. 1 Creating a Printer with the Admintool	17-6
Lesson 17. 2 Viewing Printer Settings.....	17-7
Lesson 17. 3 Using the Print Manager GUI tool for Networked Printers.....	17-10

Introduction

Printers are almost never attached to Solaris 9 servers. Companies do not spend \$50,000 for a high end SPARC server just to have it print documents. A Microsoft Windows workstation or Linux server would be a much cheaper alternative. Most Sun servers are located in server rooms and are used for tasks like hosting a Web server, working with databases, and other high end applications that need 99.99% uptime. Solaris 9 workstations have some printing needs, but with the increasing popularity of Linux as a “free workstation,” Solaris 9 workstations are starting to fall out of corporate favor.

With Sun Microsystems’ decision to not produce an Intel version of Solaris 9, printer manufacturer support might fall even further behind. The obvious result is that very few printers have Solaris print drivers. On the other hand, there is almost no printer produced in the United States that does not have a Microsoft Windows print driver.

The only reason printer support is being covered in this book is so that the book can serve as a reference for readers who are trying to pass Sun Certification exams. The easiest way to learn about printing on Solaris 9 is to purchase a very old HP printer. Solaris 9 has built-in generic print HP drivers that almost always work on HP printers (just like VGA will work with any SVGA color monitor).

The generic HP drivers installed with Solaris 9 do not support advanced features. High end HP printers require print drivers from HP’s webpage <http://www.hp.com>. These print drivers are very difficult to set up and are not necessary for the lessons covered in this chapter. If another printer is available, check for other printers that are Solaris compatible.

Some of the printers that are natively supported are:

- HP Printer**
- Reverse PostScript**
- Epson 2500**
- IBM ProPrinter**
- Qume Sprint 5**
- Daisy**
- Diablo**
- Datagraphix**
- DEC LA 100**
- DEC LN03**
- DECWriter**
- Texas Instruments 800**

Solaris 9 has expanded support for printers. Printers that have USB connectors can now run on Solaris 9. Also, Solaris 9 supports LDAP as a directory source for printers. On previous versions of Solaris the **lpsched**

process (which listens for print requests and starts the **lp** print service) was running all the time, even when a printer was not installed on the system. Now the **lpsched** process becomes active only when a printer is installed. If all the printers are removed from a system, the **lpsched** process becomes inactive.

The Print Manager GUI is much easier to use than the Admintool GUI. Remember that Sun is going to drop support for the Admintool GUI some time in the near future, so use the Print Manager GUI instead.

Printer names can not be more than 14 characters in length. The only characters allowed are **a-z**, **A-Z**, underscore (**_**) and dash (**-**). When it comes to printers, it is a good idea to make the name meaningful. Some administrators like to use names of Greek gods or Star Trek characters, such as Thor, Zeus, or Spock. Unfortunately, these names do not convey any information. A good idea is to use the location and type of printer for the name. If there an HP LaserJet 5si in building #3, second floor, use the name "BL3FL2HP5si" instead of "Thor." Most companies add "-PS" for PostScript printers, so you might have "BL3FL2HP5si-PS." If 14 characters are available, use all of them!

Printing Concepts

A print job can be printed to a local printer or to a remote printer. The printer can be set up using command line utilities or the Admintool utility. Solaris 9 also includes another GUI tool, the Print Manager.

Before we examine the printer setup tools, there are some print terms you should become familiar with.

Some Common Terms Used with Printing in Solaris 9

Print server	A system used to manage directly connected printers or remote network printers.
Print client	Any system that uses a print server.
Local printer	A printer connected to this system.
Network printer	A printer that has its own Ethernet card and IP address.
Remote printer	Any printer that is not directly connected.
Initialization	The process by which a print client prepares a printer to do a print job. This usually involves sending codes and commands to the printer.
Queuing	When too many print jobs are sent for a printer to handle all at once, the jobs are put into a "print queue." When the printer can handle the jobs, they are printed in order. This process is called "queuing."
Tracking	The print service tracks all print jobs. This allows users to cancel their own print jobs, view their print jobs, or move their print jobs between printers. The root can do these things for all users' print jobs.
Fault notification	If a print request can not be printed, the print server sends a fault notification via e-mail or console message to the print client.
Spooling space	The space used by jobs in the print queue. This usually takes 25-50 MB of disk space on the print server.

Important Directories, Files, and Commands for Solaris 9 Printing

This section gives a general overview of the directories, files, and commands that are used with printing in Solaris 9. Other sections will describe the files and directories in greater detail.

/usr/sbin Directory that holds administrator commands that in most cases can only be run by the root user on all print requests, or by individual users on their own print requests.

Files and commands in the **/usr/sbin** directory.

lpadmin The main command line program for the creation, management and deletion of printers. It allows administrators to set the default printer and perform other administration tasks.

lpfilter Used for administration of filters in LP print services.

lpforms Used for administration of forms in LP print services.

lpmove Used to move print requests from one printer to another.

lpsched Starts or restarts the LP print service.

lpshut Turns off a printer in the LP print services.

lpssystem Used to register foreign print hosts with the system.

lpusers Used to set print queue priorities.

/usr/bin The commands in the **/usr/bin** directory are typically used by regular users on their own print requests.

Files in the **/usr/sbin** directory.

lp Command line print command.

lpstat Used to cancel a print job (the root user can cancel all print jobs; regular users can cancel only their own print jobs).

lpget Used to view the current printing configuration.

lpset Used to set the printer configuration in the **/etc/printers.conf** file.

Other important files and directories are:

/usr/share/lib/terminfo
Holds all the recognized terminals and printers in its database of directories.

/var/spool/lp
The Print queue. Queued print jobs are sent here, allowing the user and the system to return to other tasks while waiting for a printer to become available.

/var/spool/lp/system/pstatus
A text file that shows if a printer is accepting print jobs and is enabled.

/var/spool/lp/requests/<hostname>

Shows the requested print jobs for a particular host.

/usr/lib/lp/model/standard

The standard printer interface program. This is a Bourne shell script.

/usr/lib/lp/netstandard

The standard printer interface program for network printers. This is also a Bourne shell script.

/usr/lib/lp/postscript

The PostScript filters.

/etc/lp

Holds configuration information about a printers installed on the system.

Files in the **/etc/lp** directory.

fd

The printer filter descriptor files.

interfaces

Contains files that point to the printer interface (LPT port).

interfaces/<printer-name>

Each printer is defined with a text file that defines the printer's characteristics.

logs

This directory is a symbolic link to the **/var/lp/logs** directory. This directory contains text files that show the activity of the **lpsched** process and all currently running print processes.

model

This directory is a symbolic link to the **/usr/lib/lp/model** directory, which contains standard interfaces to the printer standard (for local printing) and to the net standard (for network printing).

/etc/lp/printers/<printer_name>

This directory holds one subdirectory for each printer, named after that printer. These subdirectories include printer-specific configuration files and alert files.

/etc/lp/pwheels

Contains files for print cartridges and for printers with print wheels, such as daisy wheel printers.

Understanding the **/usr/share/lib/terminfo** Directory

The **/usr/share/lib/terminfo** directory holds all the supported printers and terminal devices. Under the **/usr/share/lib/terminfo** directory are subdirectories with single-letters names.

Here is what you will see if you give the command **ls /usr/share/lib/terminfo** to show these directories.

```
1 3 5 7 9 a b d f g h j l m o p r s u w y
2 4 6 8 A B c e G H i k M n P q S t v x z
```

This isn't very informative. It's more useful to look at the contents of these individual directories. Figure 17.1 shows the output from running the command `ls /usr/share/lib/terminfo/d` to list the contents of the "d" directory.

d100	dataspeed40	dialup	dmd-24	dte382
d132	dd5000	digilog	dmd-34	dtterm
d200	dec+basic	direct	dmdt80	dumb
d80	dec+low	direct800	dmdt80-w	dw
d800	decwriter	dm1520	dp3	dw1
daisy	delta	dm1521	dp3360	dw2
daisy+basic	dg	dm2500	ds40	dw3
daisy+lowres	dg6053	dm3025	ds40-2	dw4
datagraphix	diablo	dm3045	dt80	
datakit	diablo-m8	dm80	dt80-w	
datamedia2500	dialogue	dm80-w	dte	
datapoint	dialogue80	dmd	dte300s	

Figure 17.1 Output from `ls /usr/share/lib/terminfo/d`

Inside each subdirectory are the **terminfo** source files that start with the same letter. For example, Solaris 9 supports the **HP7550** printer. Inside the `/usr/share/lib/terminfo` directory is a subdirectory with the name **H**. This directory has a file named **HP7550**. Solaris 9 also supports the **vt100** terminal. Under the directory `/usr/share/lib/terminfo/v` there is a **terminfo** source file named **vt100** that is used to support **vt100** terminals.

To check if a printer or terminal devices is supported, type the command

```
find /usr/share/lib/terminfo -name <device_name>
```

For example, the command

```
find /usr/share/lib/terminfo -name vt100
```

displays the results following results:

```
/usr/share/lib/terminfo/v/vt100
```

Solaris GUI Printing Tools

Sun Solaris 9 provides two GUI tools for setting up and managing printers. The **Admintool** has been around since Solaris 2.51. The **Admintool**'s life cycle is coming to an end, and Sun Microsystems will not officially support the **Admintool** in the future. The next generation of printer configuration software is the **Print Manager**.

The **Admintool** shows the printers currently installed in the view window. Figure 17.2 is a screen shot of the **Admintool**, showing a printer named **myhp**.

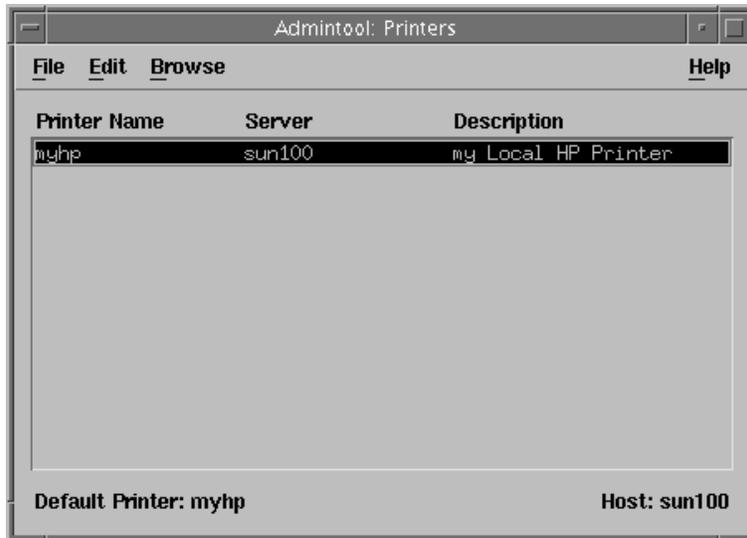


Figure 17.2 Admintool Printer View

This shows the printer's name, the server that hosts the printer and a simple text description of the printer. The Admintool can create, modify and delete local and networked printers. Remember that *networked printers* only refers to local print drivers on the system that can contact a remote printer.

The Print Manager tool can be used for administration of local and networked printers. The *networked printers* in this case refer to the print drivers located on foreign print servers. The Print Manager tool supports local printer files, NIS tables, NIS+ databases, and LDAP entries.

Lesson 17.1 Creating a Printer with the Admintool

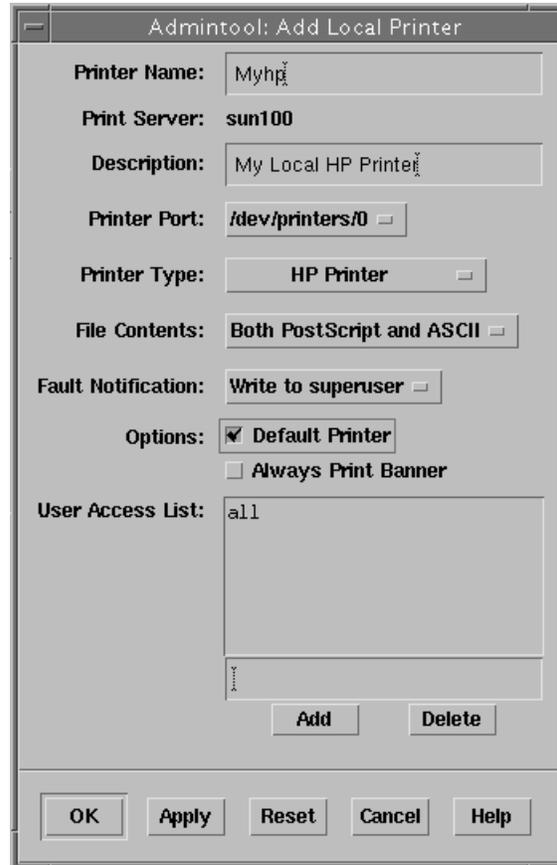
The Admintool can be used to add, modify and remove printers from a Solaris workstation. When the Admintool is started, it displays a message indicating that this tool is no longer supported by Sun and may be removed from May 2002 editions of Solaris. Please note: this book was first published on 10/2002. The default SMC 2.1 that comes with Solaris 9 does not have printer support natively built in.

1. Log in as the root user.
2. Open a Terminal window.
To open a Terminal window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Tools menu item, then left click on the Terminal icon.
3. Type the command **admintool &**
The Admintool icon is also present on the Workspace menu. To open the Workspace menu, right click anywhere in un-occupied desktop space, left click on Applications, left click on Tools, then left click on the Admintool icon.



4. Left click on Browse in the menu bar.
5. Left click on Printers

6. On the bottom of the window there should be the text "Default Printer: None"
7. Left click on the Edit menu bar item.
8. Left click on the Add menu item.
9. Left click on Local Printer...
To access a printer that is directly attached to the printer port or to a networked printer, there are two printer locations that can be used, Local Printer... and Access to Printer...
10. Configure the Add Local Printer window so it looks like the screen shot on the right.
Printer Name : Myhp
Print Server: <hostname>, (can't change)
Description: My Local HP Printer
Printer Port: /dev/printers/0
Printer Type: HP Printer
File Contents: Both PostScript and ASCII
Fault Notification: Write to superuser
Options: Default Printer,
do not select Always Print Banner
User Access list: all
11. Left click on the **OK** button
12. To exit the Admintool, left double click on the Window menu button.
The Window menu button looks like a minus sign on the top left corner of the window.
13. Open a Terminal window.
14. Type the command `lp /etc/passwd`
*The **passwd** file should be printed on the HP printer. If this does not work, try a different printer port or try a PostScript print driver.*



Possible printer ports include

```

/dev/term/a
/dev/term/b
/dev/ecpp0
/dev/printers/0 * This is the most likely port.

```

Lesson 17.2 Viewing Printer Settings

In this lesson readers examine the various files and directories that were created when the printer **Myhp** was created in the previous lesson. The only real way to understand how the print system works in Solaris 9 is to play around the various directories and files.

1. Log in as the root user
2. Open a Terminal window.
3. Type the command `cd /etc/lp/printers`
4. Type the command `ls`
*In the **/etc/lp/printers** directory there should be a sub-directory named **myhp***
5. Type the command `cd myhp`
6. Type the command `pwd`
*The current directory should be **/etc/lp/printers/myhp***
7. Type the command `ls`

The next couple of steps examine the files in the `/etc/lp/printers/myhp` directory.

8. Type the command `cat comment`
This shows the text comment associated with the printer.
9. Type the command `cat users.deny`
To deny a user the ability to print, the system administrator can enter that person's username in the `users.deny` file. This file is initially empty.
10. Type the command `cat configuration`
The output will look something like this:
Banner: optional
Content types: postscript,simple
Device: /dev/printers/0
Interface: /usr/lib/lp/model/standard
Printer type: hplaser
Modules:
Options:
The configuration file is a very critical file that sets the printer's configuration. If there is a problem with a printer, check this file to see its settings. These settings include:
 - **Device:** `/dev/<raw-device-file>`. In this example, this is **Device: /dev/printers/0** If the printer does not print, check this setting on other similar SPARC workstations. The device file for the printer port or serial port to which the printer is attached should be the same in your setup as in other setups.
 - **Interface:** `/usr/lib/lp/model/standard` or
 - **Interface:** `/usr/lib/lp/model/netstandard`*One of the interface files should be present for this printer.*
11. Type the command `cd /etc/lp`
12. Type the command `ls classes`
This directory only contains files if the printer is in a class of printers A printer class is similar to Microsoft's pool of printers.
13. Type the command `ls fd`
The `fd` directory contains printer filters. A printer filter reconfigures a print job to match a specific print format, such as HP's PCL language or PostScript.
14. Type the command `cd /etc/lp/interfaces`
This directory contains a printer interface program for each installed printer.
15. Type the command `more myhp`
As shown by the `more` command, this file contains global variables. If you encounter problems, check these variables against other functional printer files:
 - TMPDIR=/tmp** - The system's temporary file system. It should be `/tmp`
 - SPOOLDIR=/usr/spool/lp** - Where print jobs are spooled when the system is waiting for the printer or print server to take the job. Make sure that the `/usr` file system still has some space available.
 - TERMINFO = /usr/lib/terminfo** - Where the Solaris 9 operating system gets printer and terminal device information.
 - CHARSETDIR** - The character sets used with the printer.*If there are printing problems, look for error messages in the `/etc/lp/interfaces/<name_of_printer>` file (in this case, this file is `/etc/lp/interfaces/myhp`), such as like **unknown printer/interface failure**.
*Look at the script lines and comment lines in this file. Compared to most files, this file has excellent comments to help you understand it. As you read the file, ask the questions: What programs are being started? What directories are being used? Do the programs exist? Are the directories present? What are the permissions on the directories?**
16. Type the command `cd /etc/lp`
17. Type the command `cd forms`
18. Type the command `ls`
The `forms` directory contains printer forms, if they exist.
19. Type the command `cd /etc/lp`

20. Type the command `cat filter.table`
Solaris 9 stores information about print filters. A print filter converts a print job, for example, from non-PostScript to PostScript.
21. Type the command `cd /etc/lp`
22. Type the command `ls -l logs`
Notice that the `/etc/lp/logs` are symbolically linked to the directory `/var/lp/logs`
23. Type the command `cd /etc/lp/logs`
24. Type the command `pwd`
Notice that the present working directory is `/var/lp/logs`
25. Type the command `more lpsched`
This indicates when the `lpsched` process started. The `lpsched` process starts the Print service. It includes information that answers the following questions for each print job: When was the print job started? Did the `lpsched` start? Did the Print service start and stop? Was the print job successful?
26. Type the command `more requests`
This file may not exist. If it does, it is very difficult to read. The main point of looking at it is to see information about each print request that has been logged.

Here is a sample of this command's output. Some of the lines explained below do not appear in this sample.

```

= myhp-1, uid 0, gid 1, size 1163, Thu Sep 26 18:36:45 MDT 2002
z myhp
C 1
D myhp
F /etc/passwd
P 20
T /etc/passwd
t simple
U root
s 0x0010
v 2

```

The letters at the start of lines in this file indicate:

C Number of copies
D The printer's destination or the class destination; can also be the word **any**
f name of form
F Name of the file printed
H Resume, immediate, hold
N<M> Successful print job notified via email
N<W> Successful print job notified via message on a terminal connection

Optional flags (portrait printing,landscape printing)

- p* Priority of the print request
P A list of the pages printed
r Raw processing of a print job
S Special character set to use
T The banner page title
t Type of file
U Username of the person who submitted the print request
x Slow filer used for the print request
y Specific printer models that the filters should use
z The printer's name
27. Type the command `cd /etc/lp/model`

28. Type the command `pwd`

The directory `/etc/lp/model` is symbolically linked to the `/usr/lib/lp/model` directory. There should only be two files in this directory, `netstandard` and `standard`. They are used for network and local printers, respectively.

Lesson 17.3 Using the Print Manager GUI tool for Networked Printers

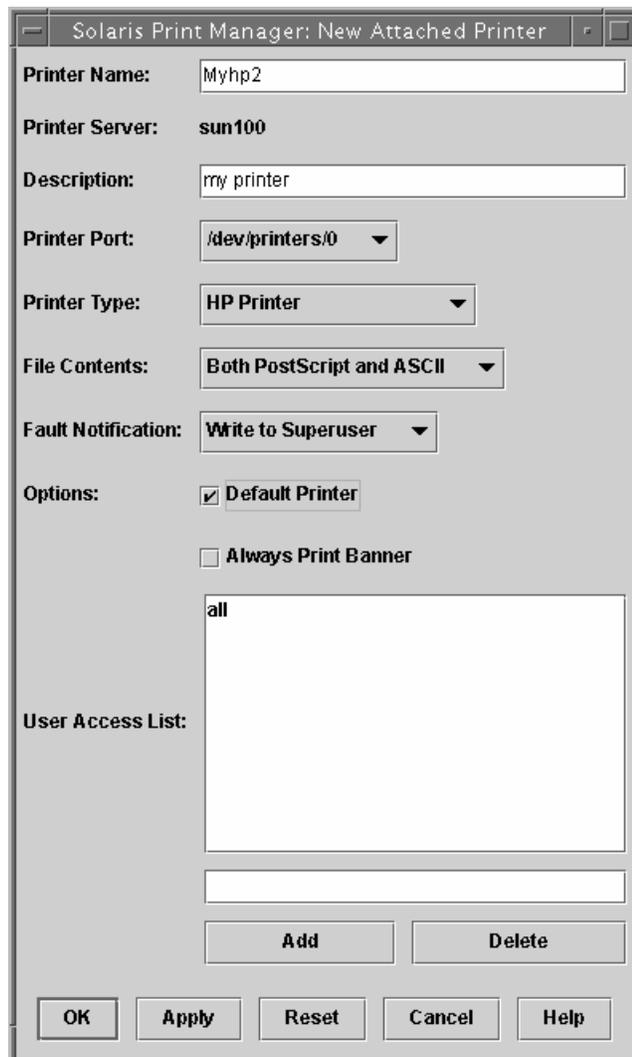
Printers can be managed with a GUI tool know as the `printmgr`. In this lesson, readers will use `printmgr` to create a local HP printer.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command `/usr/sadm/admin/bin/printmgr &`
A small popup window should appear that looks like this:
4. Left click on the **OK** button.
5. Left click on the **Printer** menu item..
6. Select **New Attached** printer...



7. Select **Local Printer...** Enter or choose the information shown below.

PrinterName	[Myhp2]
PrintServer:	[sun100] (can not change)
Description:	[my printer]
Printer Port:	[/dev/printers/0]
Printer Type:	[HP Printer]
File Contents:	[Both PostScript and ASCII]
Fault Notification	[Write to Superuser]
Options:	<input checked="" type="checkbox"/> [Default Printer]
User Access List:	[all]



8. Click on the **Add** button at the bottom of the window.
9. Click on **OK**
10. Type the command `lp -p myhp2 /etc/passwd`
This command prints the text file `/etc/passwd` to the default printer.

Command Line Print Management Tools

There are various command line tools that can be used to set up printers in Solaris 9. The reason there are so many different printer-related command line tools is so that older scripts or scripts written for different systems can be run on Solaris 9 without further modification. The following commands can work the Solaris 9 operating system.

lpadmin The **lpadmin** command is the primary command used to add, monitor and remove printers within Solaris 9. The **lpadmin** command is also useful in changing the system default printer setting.

accept/reject Continue to accept print requests, or deny any further print requests. If a printer needs maintenance, the command **reject** is typed, and the printer can be serviced. When the printer is ready to accept print jobs, the command **accept** is used to start the printer again.

lpstat Shows the status of a printer.

lpmove Move print requests between printers. This is very useful if a printer becomes flooded with print requests. The **lpmove** command can be used to move the print requests to a similar printer.

enable/disable When disabled, print requests can still come into the print spool area, but the requested documents will not be printed. The **disable** command is also a maintenance command that lets a system administrator hold users' print jobs in the spool area until the enable command is typed.

Alternate Command Line Print Commands

Solaris 9 supports two different commands to print from the command line. The **lp** command is the native System V line print command. This is the most common command line print command is used in most industry scripts. The command `lp <text_file_name>` prints the text file to the default printer.

The **lpr** print command is used for BSD compatibility, so that BSD scripts and BSD users can print without any difficulty.

`/usr/bin/lp` the standard print command used to print text files

For scripts that reference and use the BSD print arguments with the **lp** command, the following `lp` print location can be used.

`/usr/ucb/lp` old BSD compatible print command to print text files

The **lp** and **lpr** print commands will automatically try to use the default printer. If a user needs to print to another printer, one of the following commands can be tried to print to a non-default printer.

One style of print command is the atomic print command. This style uses the `-d` option to specify a different printer that is known to this system.

```
/usr/bin/lp -d printerA /etc/passwd
```

or for BSD compatibility, the following command can be used to specify a non-default printer.

```
/usr/ucb/lpr -P printerA /etc/passwd
```

To specify a printer on a remote system, try the POSIX style print command

```
/bin/lp -d server1:printerA /etc/passwd
```

or the BSD version of this command:

```
/usr/ucb/lp -P server1:printerA /etc/passwd
```

How Solaris Finds the Default Printer

If a printer is not defined by the user, Solaris 9 looks in the following places to determine the default printer.

1. The command line
2. The system **PRINTER** or **LPDEST** variables
3. The user's **\$HOME/.printer** file
4. The **/etc/printers.conf** file
5. The NIS map **printers.conf.byname**

If none of these places provides the necessary information, Solaris 9 will not be able to find the printer.

Key Points to Remember

Because printers are rarely attached to a server, printers are not a big issue in Solaris 9. Most system administrators do not even install print drivers on Solaris servers. This chapter has at least given the reader an exposure to printer management under Solaris. Solaris workstations do use printers so, it is a good idea to have some idea of how printers work with Solaris. The key point is to not to spend too much time learning how printers work on Solaris. Over 95% of Solaris servers in this world do not interact with a printer.

Chapter 18 Backup and Restore

Lessons in This Chapter

Lesson 18.1 Using the <code>ufsdump</code> and <code>ufsrestore</code> Commands	18-7
Lesson 18.2 Using <code>ufsrestore</code> in Interactive Mode	18-9
Lesson 18.3 Using the <code>tar</code> Command	18-11
Lesson 18.4 Using the <code>dd</code> Command	18-16
Lesson 18.5 Using <code>dd</code> to Copy a Hard Drive	18-17

Introduction

This chapter demonstrates various file system backup and restore commands. These backup and restore commands will require a second hard drive as the backup device. The hard drive can be of almost any size. A floppy diskette can also be used to help demonstrate backups and restores. The only problem is that the Solaris 9 operating system has very bad command line support for floppy diskettes. These commands are cryptic and there are times that the operating system does not recognize that a floppy has been formatted, added, changed or removed (even when the `volcheck` and `volmgt` commands are used.)

If the reader has a computer without a second hard drive on each workstation, use the floppy. Just make sure to practice with the floppy commands extensively so it's possible to get out of messy situation.

Some common commands to work with a floppy diskette are:

- **fdformat -H**
-H Sets the format at high density.
- **fdformat -U -H**
-U Unmounts the floppy diskette.
-H Sets the format at high density.
- **newfs /vol/dev/rdiskette0/unnamed_floppy**
Creates a new file system on the floppy diskette (provided the floppy label is `unnamed_floppy`).
- **newfs /vol/dev/rdiskette0/<disklabel>**
Creates a new file system on a floppy diskette with a custom label.
- **volcheck**
This command is used to check each device that has removable volumes. This includes floppy diskettes, CDROMs and DVDs.

If the `volcheck` command does not recognize a change in the floppy diskette, try these commands:

1. Type `/etc/init.d/volmgt stop`
2. Type `umount -F /floppy/floppy0`

3. Type `rm -rf /floppy`
4. Type `/etc/init.d/volmgt start`
5. Type `volcheck` again

This chapter covers the use of the `ufsdump`, `ufsrestore` and `tar` command for backups and restores of files, directories and slices. These lessons use an extra hard drive or floppy drive to simulate a tape drive. Unfortunately, a tape drive and a disk drive work in different ways, so the simulation is not perfect. Most SCSI tape drives cost hundreds if not thousands of dollars, so a second hard drive is a good teaching substitute.

Before starting each lesson, make sure to format several floppy diskettes so that they can be used to simulate tapes. An extra hard drive can act like a tape drive in raw mode with `ufsdump` and `ufsrestore` commands. It is impossible to use the `mt` command with a hard drive, so that part of the simulation can not be completed.

The `ufsdump` and `ufsrestore` Commands

The `ufsdump` and `ufsrestore` commands are used to back up an entire file system. The `ufsdump` command is typically used to produce full and incremental backups to a tape drive or another storage device.

A *full* backup copies all the files to the tape drive or other backup media. An *incremental* backup copies only the files that have changed since the last full backup or incremental backup with a lower number. A full backup is represented by the number zero (0). An incremental backup is represented by the numbers one (1) through nine (9).

- Full backup – (0) - Copies all files, regardless of when the file was created.
- Incremental backup - (1) – Copies any files that have been added or modified since the last full backup.
- Incremental backup - (2) Copies any files that have been added or modified since the incremental backup (1).
- Incremental backup – (x) Copies any files that have been added or modified from any lower number backup (x -1).

The situations shown below give examples of using backup numbers.

Situation 1

A company has the following backup schedule:

SUN	MON	TUE	WED	THR	FRI	SAT
0	1	1	1	1	1	1

Here, a full backup (0) is done on Sundays, and an incremental backup (1) is done on all other days. Every day, a new tape is put in the tape drive.

Let's say that one Sunday, the system hard drive crashes fifteen minutes after the full backup is performed. In this case, the system administrator would restore the system files from the Sunday backup tape. The file system has been fully restored to its last good condition.

Advantage of this backup schedule The tape system has fast system restores. Just restore the full level (0) tape and the daily level (1) tape.

Disadvantage of this tape system This system wastes a lot of tapes. Any file created or modified on Monday, Tuesday, Wednesday or any other day is repeatedly copied onto the level (1) tapes.

Situation 2

A company has the same backup schedule as above:

SUN	MON	TUE	WED	THR	FRI	SAT
0	1	1	1	1	1	1

As above, a full backup is performed each Sunday, and an incremental backup (1) is done on all other days. Every day, a new tape is put in the tape drive.

Let's say that on Thursday the server crashes after the incremental tape backup is done. The system administrator installs a new hard drive. The system administrator now performs a restore from the full backup tape that was created on Sunday. He or she then pulls out the Thursday tape and restores all files from that tape. Remember, the incremental tape only has files on it that have been modified and created since the level (0) backup on Sunday. Now the system has been fully restored.

Advantage of this backup schedule - The tape system has fast system restores. Just restore the full level (0) tape and the daily level (1) tape.

Disadvantage of this tape system - This system wastes a lot of tapes. Any file created or modified on Monday, Tuesday, Wednesday or any other day is repeatedly copied onto the level (1) tapes.

Situation 3

A company has the following backup schedule:

SUN	MON	TUE	WED	THR	FRI	SAT
0	1	2	3	4	5	6

Each day a new tape is put into the tape drive.

This backup strategy is a bit more complicated:

1. A full backup of the system performed on Sunday.
2. Each day an incremental backup is performed on the system with a new tape. Take a look at the numbers for the other days (1-6). Each day, all files changed *since the previous day's backup* are copied to a new incremental backup tape.

Let's say that on Thursday the server crashes after the backup tape was created. The system administrator installs a new hard drive. The restore process with this setup is also more complicated. The system administrator must now restore files from the following sources, in the order shown:

1. Full restore from the full backup tape that was created on Sunday.

2. Incremental restore from the Monday level (**1**) tape - restores files created or modified on Monday.
3. Incremental restore from the Tuesday level (**2**) tape. restores files created or modified on Tuesday.
4. Incremental restore from the Wednesday level (**3**) tape - restores files created or modified on Wednesday.
5. Incremental restore from the Thursday level (**4**) tape.- restores files created or modified on Thursday.

Now the system has been fully restored.

Advantage of this backup schedule - This schedule saves tape space. Any file created on a week day is only saved on one tape. The Thursday tape will not copy a file created or modified on Monday or Tuesday. The Thursday tape only copies files created or modified since the Wednesday tape.

Disadvantage of this tape system – This schedule is rather time consuming for the system administrator to restore all those tapes.

An obvious question would be why any system administrator would want to perform any incremental backup other than a level (**1**) incremental backup.

If an administrator makes a level (**4**) incremental backup and then needs to restore a system, the full backup tape would have to be restored first. Then, the system administrator would have to restore all the incremental tapes from level (**1**) to level (**4**), in that order. Remember that in Situation 2, the system administrator only had to restore the Sunday level (**0**) tape and the Thursday level (**1**) tape.

The reason that companies usually want to have more than a level (**1**) backup is due to economics. Most companies with hundreds of servers can not afford to copy a file onto several tapes. A level (**1**) backup performed every day would copy every file created or modified since Sunday onto every weekday tape. With a large corporation that would become very cost prohibitive. If a company uses the different level incremental backups shown in Situation 3, a file created on Monday would not be saved on the Tuesday tape, the Wednesday tape and the Thursday tape. So, most companies choose to use the strategy in Situation 3.

Given the strategy in Situation 3, the easiest way to understand how to fully restore a system is:

1. Restore the last full backup tape.
2. Starting with the lowest level incremental tape; restore all higher level incremental backup tapes until the current tape is reached

For example:

If the last full level (**0**) backup tape was created on Friday (or any day...) restore that tape first. Next, restore the next highest level tape you can find. Let's say that you find a level (**1**) tape created on Monday. Restore that tape. Then, continue to restore higher numbered tapes until the system is fully restored.

Quick Tip

- Occasionally, people with a background in databases have the mistaken belief that the terms *incremental copy* and *incremental restore* mean that *part of a file* is copied or restored. This is wrong! The tape system either copies a file in its entire form or it does not copy that file at all.
- An incremental backup and restore only references the fact that a file is copied or not copied. If a user created a text file with three lines of text on Sunday, that file would be copied to the tape. If the user added some more lines of text to the file on Wednesday, the file would be copied in its entire form again.
- The backup and restore utilities provided with Sun Solaris 9 do not copy file fragments or parts of files.

Tape Drive Locations

When you back up or restore files, you will need to specify what tape unit to use. Here are the standard locations:

<code>/dev/rmt/0</code>	The first tape drive on the system
<code>/dev/rmt/1</code>	The second tape drive on the system
<code>/dev/rmt/2</code>	The third tape drive on the system
<code>hostx:/dev/rmt/1</code>	The second tape drive on another server, which is named "hostx"

The `ufsdump` Command

The `ufsdump` command is used to produce full and incremental backups of an entire file system or a large directory structure. It is not intended to be used to copy just a couple of files or just a single directory. The `ufsdump` utility can be used to copy selected files and directories if it is absolutely necessary, but it is a poor tool for such a job.

The `ufsdump` command supports the following options:

`ufsdump < options >`

- f** Specifies a tape device other than the default `/dev/rmt/0` tape drive.
- o** (small **O**) Used to take a tape offline, rewind the tape, and eject the tape.
- l** (small **L**) Used to load a tape into an autoloading tape device
- s** Tells `ufsdump` to estimate the size of the backup (does not perform a backup).
- u** Updates the `/etc/dumpdates` file. This file saves the dump level and the date the dump level was performed. This is a handy reference file that is useful to determine what backups were performed on what dates. If the **u** option is not used, the `/etc/dumpdates` file is not updated or saved.

v Verifies the backup against the original files. Please note: if the original files change during the backup process, false error messages could be reported by the **ufsdump** command.

Examples of the **ufsdump** Command

ufsdump 0f /dev/rmt/1 /opt

This command creates a full backup (**0**) using the second tape drive (**f /dev/rmt/1**) of the **/opt** directory (**/opt**).

ufsdump 0v /opt

This command creates a full backup (**0**) with verification (**v**) of the **/opt** directory. If no tape drive is specified with the (**f**) option, the first tape drive (**/dev/rmt/0**) is used as a default.

ufsdump S /opt

This command tells **ufsdump** to report the size of the dump file (**S**) needed to back up the **/opt** directory. No actual backup would be performed.

ufsdump 0l /opt (*zero little L/opt*)

This command creates a full backup (**0**) and loads a tape into an autoloading tape device **l** (small **L**) of the **/opt** directory (**/opt**).

The **/etc/dumpdates** file has the following format:

File System Backed Up	Backup Level	Day	Month	Day	Time	Year
/dev/rdisk/c0t3d0s4	2	Mon	May	24	13:23:14	2002

Types of Tapes

Some standard tapes, and their sizes, are:

½" reel tape	140 MB
¼" QIC cartridge tape	8 GB
4mm DAT	24 GB
8mm DAT	40 GB
DLT ½" tape	70 GB

The **ufsrestore** command

The **ufsrestore** command restores files from tapes that the **ufsdump** command created. The files can be restored individually, or an entire file system can be restored automatically. To restore a file system, the system administrator must create a mount point and then **cd** to that mount point. Next, he or she has to run the **ufsrestore** command to restore the file system.

The **ufsrestore** command restores files into the directory that the command was run from. To restore the root (**/**) file system the root file system must not be actively running. Basically, the system must be brought down and the replacement hard drive that is to hold the root file system must be mounted remotely. This is explained in a later lesson.

`ufsrestore < options > /raw_device_name_of_the_backup_device`

`< options >`

- f** Specifies the tape device or backup device.
- i** Starts an interactive restore.
- r** Restores the entire file system.
- t** Lists the contents of the tape or backup device.
- x** Restores only the files specified on the command line (add file names to the end of the command line)

`/raw_device_name_of_the_backup_device`

This argument looks like:

`/dev/rmt/0` (first tape drive on the system)
`/dev/rmt/3` (fourth tape drive on the system)
`/dev/rdisk/c0t1d0s4` (slice 4 on the second hard drive)

As an example, the command

`ufsrestore rf /dev/rmt/0`

would restore the file system from the first tape drive on the system, to the current mount point.

Lesson 18.1 Using the `ufsdump` and `ufsrestore` Commands

In this lesson floppy diskettes or a hard drive will be used to simulate the use of different tapes. The `ufsdump` and `ufsrestore` commands are the best commands to back up an entire partition or slice.

Notes: (1) You can use either a hard drive slice or floppy diskettes to perform the backup and restore operations in this lesson. Steps that involve floppies begin with the marker [**Floppy step**]. If you use a hard drive slice, skip all steps that involve floppies.

(2) The previous contents of all floppy diskettes used in this lesson will be overwritten. Only use diskettes that have no data that you want to keep!

(3) The previous contents of any hard disk slice used in this lesson will be overwritten. Only use a hard disk slice that has no data that you want to keep!

1. Log in as the root user.
2. Open a Terminal window.
To open a Terminal window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Tools menu item, then left click on the Terminal icon.
3. [**Floppy step**] Insert a floppy in the floppy diskette drive.
4. [**Floppy step**] Type the command `fdformat -U -H`
A message should appear, saying:
Formatting 1.44 MB in /vol/dev/rdiskette0/unnamed_floppy
Press Return to start formatting floppy.
Press the return key.
5. [**Floppy step**] Type the command `newfs /vol/dev/rdiskette0/unknown_format`
A message should appear, saying:

newfs: construct a new file system /vol/dev/rdiskette0/unknown_format (y/n)? y

Type **y** and press the return key.

6. Type the command **init 6**
*The command **init 6** reboots the workstation. For this lesson, It is important that the backup example start from a fresh system that just went through a reboot. This is to ensure there are no unusual hung processes or problems with the file systems.*
7. Log in as the root user.
8. Open a Terminal window.
9. [**Floppy step**] Type the command **volcheck**
*The **volcheck** command checks all removable devices (floppy drive, CDROM, DVD, etc.) for a new disk. The floppy drive light should come on, and a directory named **/floppy** will automatically be created as a mount point for the floppy diskette.*
**FOLLOW THE NEXT FOUR STEPS IF THE FLOPPY CAN NOT BE SEEN.
IF THE FLOPPY CAN BE SEEN, SKIP TO STEP 14.**
10. [**Floppy step**] Type the command **/etc/init.d/volmgt stop**
11. [**Floppy step**] Type the command **rm -rf /floppy**
*The **/floppy** directory might not exist, so ignore any error messages from this command.*
12. [**Floppy step**] Type the command **/etc/init.d/volmgt start**
13. [**Floppy step**] Type the command **volcheck**
*If the **/floppy** directory does not exist, type the command **volcheck** again.*
14. [**Floppy step**] After the floppy is recognized, type the command **cd /floppy**
15. [**Floppy step**] Use the **ls** command to make sure there is a directory named **/floppy/floppy0** exists. If this directory does not exist, go back to step 10.
16. Type the command **mkdir /slice6**
*The directory **/slice6** might already exist, so you might get an error message. Do not worry if this happens.*
17. Type the command **mount /dev/dsk/c0t0d0s6 /slice6**
*If the directory **/slice6** is already mounted, type the command **mount | grep slice6** and make sure it is mounted on the 6th slice of the first disk.*
18. Type the command **cd /slice6**
19. Type the command **ls -l**
*If any files or directories exist in **/slice6**, copy them to another source before proceeding.*
20. After copying any files or directory from **/slice6**, delete everything except for a directory with the name **lost+found**. This directory exists on all UFS file systems and should not be deleted.
21. Type the command
echo "file created for backup demo" > mysavfile
22. Type the command **ls -l (-l is "little L")**
*Make sure there is a file named **mysavfile** in the **/slice6** directory.*
23. [**Floppy step**] For systems that are using the floppy diskette as the backup device, type the following command:
ufsdump 0f /vol/dev/aliases/floppy0 /dev/rdisk/c0t0d0s6
A message like the following should appear on the screen:
ufsdump 0f /vol/dev/aliases/floppy0 /dev/rdisk/c0t0d0s6
DUMP: Writing 32 Kilobyte records
DUMP: Date of this level 0 dump: Tue 10 Sep 2002 09:27:08 PM MDT
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rdisk/c0t0d0s6 (sun100:/slice6) to /vol/dev/aliases/floppy0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Estimated 112 blocks (56KB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: 62 blocks (31KB) on 1 volume at 7 KB/sec
DUMP: DUMP IS DONE

24. For systems that are using the second hard drive as the backup device, find an unused slice on the second hard drive with at least 50 MB of free space and type the following command:

```
ufsdump 0f /dev/rdisk/c#t#d#s# /dev/rdisk/c0t0d0s6
```

Where it says **c#t#d#s#** above, substitute the Logical Device Name for your chosen slice. The author's system had no data in slice 7 on the second hard drive, so the author would use **c0t2d0s7**. You can use the **format** command to find your system's hard drive Logical Device Names.

A message like the following should appear on the screen:

```
# ufsdump 0f /dev/rdisk/c0t2d0s7 /dev/rdisk/c0t0d0s6  
DUMP: Writing 32 Kilobyte records  
DUMP: Date of this level 0 dump: Thu Sep 26 22:02:03 2002  
DUMP: Date of last level 0 dump: the epoch  
DUMP: Dumping /dev/rdisk/c0t0d0s6 (sun100:/slice6) to /dev/rdisk/c0t0d0s7.  
DUMP: Mapping (Pass I) [regular files]  
DUMP: Mapping (Pass II) [directories]  
DUMP: Estimated 118 blocks (59KB).  
DUMP: Dumping (Pass III) [directories]  
DUMP: Dumping (Pass IV) [regular files]  
DUMP: 62 blocks (31KB) on 1 volume at 344 KB/sec  
DUMP: DUMP IS DONE
```

25. Type the command **cd /**
*Now that slice 6 has been backed up, we will wipe out the contents of that slice. This will let us use **ufsrestore** to restore those contents later in this lesson. This step starts the process of wiping out the contents of slice 6.*
26. Type the command **umount /slice6**
27. Type the command **newfs /dev/rdisk/c0t0d0s6**
*The **newfs** command creates a brand new file system on slice6, wiping out any data on that slice.*
28. Type the command **mount /dev/dsk/c0t0d0s6 /slice6**
29. Type the command **cd /slice6**
30. [Floppy step]. For systems that only have a floppy disk, type the command **ufsrestore rf /vol/dev/aliases/floppy0**
*If a message asks for the next volume, type 1 (the number "1"). There might be an error message "Warning: /lost+found exists." Ignore this message.
Congratulations! You have backed up a file system to a media (a floppy drive this time) and then restored it*
31. For systems that are using the second hard drive as the backup media, type the command **ufsrestore rf /dev/rdisk/c0t2d0s7**
*If a message asks for the next volume, type 1. There might be an error message "Warning: /lost+found exists." Ignore this message.
Congratulations! You have backed up a file system to a media (a hard drive this time) and restored it!*
32. Type the command **ls /slice6**
The contents that you backed up and wiped out have now been restored.

DO NOT CLOSE DOWN THE TERMINAL WINDOW OR LOG OFF!

Lesson 18.2 Using ufsrestore in Interactive Mode

Lesson 18.2 is a carry over lesson from Lesson 18.1. Only perform Lesson 18.2 immediately after successfully performing Lesson 18.1.

In this lesson the reader will perform an interactive restore of the backed up file **mysavefile** to slice6 on the first hard drive.

1. Type the command **volcheck**
2. Type the command **cd /slice6**
3. Type the command **rm mysavefile**
4. [**Floppy step**] For systems that are using a floppy diskette as the backup media, type the command
ufsrestore if /vol/dev/aliases/floppy0
5. For systems that are using a second hard drive as the backup media, type the command
ufsrestore if /dev/rdisk/c0t2d0s7
6. At the **ufsrestore>** prompt, type **help**
*The help command shows the various commands that can be run during the interactive session of the **ufsdump** command.*
7. At the **ufsrestore>** prompt, type **ls**
*The **ls** command in **ufsrestore** shows the contents of the backup media.*
8. At the **ufsrestore>** prompt, type **add mysavefile**
*This command adds the file **mysavefile** to the list of files that will be extracted. The file is not extracted now, just marked to be extracted. You can use the **add** command as many times as you like, to add more than one file. Do this before you run the **extract** command.*
9. At the **ufsrestore>** prompt, type **extract**
*If there is a question about the volume number, type in **1** (one) as the volume number.*
10. At the **ufsrestore>** prompt, type **quit**
11. Type the command **ls /slice6**
*This will show that **mysavefile** has been restored from the backup to slice 6.*

The tar command

The **tar** command is used to copy and restore directories and files into and out of what is known as a *tar file* or to a tape drive. The **tar** command does have a disadvantage in that it can only back up and restore to a single tape drive. Just as Windows zip programs create files called **<name>.zip**, the **tar** command produces files called **<name>.tar**.

The **tar** command uses the following arguments:

```
tar < options > <backup_device> <target>
```

```
tar < options > <filename.tar> <files_or_directories_to_backup>
```

```
tar < options > <raw_device_name> <files_or_directories_to_backup>
```

< options >

- c** Copy files to the tar file
- x** Extract files from the tar file
- t** Show the contents of the tar file
- v** Show verbose output

<backup_device>

The **tar** command can work with a file with a **.tar** extension. The **tar** command will also work if it is given the raw device name of a backup device like

```
/dev/rmt/0
```

or

```
/dev/rdisk/c0t2d0s4.
```

<target>

The full or relative path name of the files and directories you want to archive with the **tar** command. Understand that when the file and directories are restored, they will have the same directory structure as when the tar file was saved.

For example, if a system administrator types the command

```
tar -cvf mytarfile.tar /opt
```

the **tar** command will save the entire **/opt** directory structure as an *absolute path name* in the **mytarfile.tar** file.

When a system administrator types the command

```
tar -xvf mytarfile.tar
```

a directory named **/opt** will be created, and all the files and subdirectories of the **/opt** directory will be restored.

Let's take a look at another example. Say that the system administrator types the commands

```
cd /  
tar -cvf mytarfile.tar opt
```

Take special note that here, the **opt** directory was not specified with a front slash (/) in front of the opt directory. This means that it was specified as a *relative path name*.

In this case, the **tar** command will still save the entire **/opt** directory structure in the **mytarfile.tar** file. However, when the directory structure is restored, it will be restored *relative to the current working directory*, and not to the absolute path name **/opt**.

So, when the system administrator types the following commands

```
mkdir /abc  
cd /abc  
tar -xvf mytarfile.tar
```

a directory named **/abc/opt** will be created, and all files and subdirectories of the **/opt** directory will be restored to *that* directory, rather than to **/opt**.

Lesson 18.3 Using the tar Command

In this lesson the **tar** command will be used to back up a simple text file. The contents of the tar file named **mytarfile.tar** will be examined. Later, the text file will be extracted out of the tar file and returned to its original location. Also, the contents of the **/etc** directory will be saved to the tar file named **etcsave.tar** and extracted to a different directory, **/abc**.

1. Log in as the root user.
2. Open a Terminal window.

3. Type the command `cd /`
4. Type the command.
`echo "this is a file to be saved" > mytextfile2`
This command creates a text file named `mytextfile2` with the line "this is a file to be saved" in it.
5. Type the command `ls`
This command shows all the files in the current directory. The file `mytextfile2` should be listed.
6. Type the command `tar -cvf mytarfile.tar mytextfile2`
This command makes a tar file named `mytarfile.tar` that contains the text file `mytextfile2`.
7. Type the command `rm mytextfile2`
This command removes the text file `mytextfile2` from the system.
8. Type the command `ls`
Notice that the file `mytextfile2` has been removed from the directory.
9. Type the command `tar -xvf mytarfile.tar`
This command extracts all the files from `mytarfile.tar` file.
10. Type the command `ls`
The file `mytextfile2` has been restored from the tar file.
11. Type the command `tar -cvf etcsave.tar etc`
This command saves the entire `/etc` directory structure in the file `.etcsave.tar`
12. Type the command `tar -tvf etcsave.tar`
This command shows the contents of a tar file.
13. Type the command `mkdir /abc`
This command creates an ordinary directory named `/abc`.
14. Type the command `cd /abc`
15. Type the command `tar -xvf etcsave.tar`
This command extracts all the files from `etcsave.tar` file to the current working directory, `/abc`
16. Type the command `ls`
Explore the `/abc` directory. It now contains a copy of the `/etc` directory.

Compressing, Viewing and Uncompressing Files

When a `.tar` file is created, files are copied into it in compressed form. However, it is very common to compress `.tar` files again before putting them on a website or FTP site. Why would anyone want to compress an already compressed file? Because by using two compression utilities on a file it is possible to further reduce its size. This is particularly important for users who transfer files over a 56K (or slower) modem connection. Also, compression utilities can copy many files (and many directories) into one file. That way, users can download a single file, rather than many separate files.

Solaris 9 features various compression utilities. The most commonly used utilities are:

compress/uncompress	Create and use <code>.Z</code> files. These utilities are not universally used with all operating systems.
zcat	Used to uncompress <code>.Z</code> files. This utility is used among different operating systems.
zip	Creates and uses the famous <code>.zip</code> files. This utility is very widely distributed among many different types of operating systems.
gzip	An open source version of the zip utility that is very commonly distributed. It produces files with the <code>.gz</code> file extension.

All of these utilities can copy an image of files and directories into one compressed file. This compressed file can be easily copied from one location to another via through utilities or media such as FTP, CDROMs and floppies. Once at their destination the compressed file can be uncompressed into its original source files and directories.

The zip and unzip Utilities

The **zip** utility compresses and copies the contents of files and directories into a single **.zip** file. It can also copy files to a file system or tape drive, instead of to a **.zip** file. The original source files and directories are not changed. A **.zip** file can then be copied to any system that supports the **zip** utility, such as UNIX, MS-DOS, VMS, OS/2, Microsoft Windows and a host of other operating systems. A collection of files and directories can be compressed on a UNIX server and then be uncompressed on any other operating system that has includes a **zip** program.

Notice that there are some cross-platform issues regarding maximum filename lengths. Some operating systems and programs (such as MS-DOS and MS-DOS zip file programs) limit the length of filenames. For example, MS-DOS has the "8.3" limit, in which filenames are limited to eight characters, a period, and then three more characters. So, if you use an MS-DOS zip program to compress a file called **March Accounts.txt**, the filename inside the zip file may be shortened to something like **MARCHA~1.TXT**. Notice that this filename fits the "8.3" limit, and that all characters in it have been changed to upper case. When the file is uncompressed, it may be restored under the truncated name **MARCHA~1.TXT**.

Names of directory structures can also cause problems. UNIX-based operating systems (such as Solaris) use the forward slash (/) between directory names. Some operating systems, such as Microsoft Windows, use the backward slash (\) instead. With some older zip utilities, this difference can cause problems. Also, some operating systems, such as Microsoft Windows, limit the full path name of a file (including all slashes between directory names) to 64 characters. It's a good idea to make sure that you keep all path names for files under 64 total characters if you will be transferring them to another operating system.

The **zip** utility saves critical file information about the files that it compresses, such as name, path, date, and last time the file was modified.

A system administrator can use the **zip** utility to uncompress **.zip** files. There is another command named **unzip** that can also be used to decompress **.zip** files.

The **zip** utility supports the following options:

```
zip <options> <filename.zip> <files to be compressed>
```

<options>

- b (/directory)** Use a temporary directory to create or modify a **.zip** file. This is useful if the source file system can not handle the extra disk space requirements for updating a **.zip** file.
- c** Add comments about a file.
- d** Delete a file from a zip archive.
- D** Do not create directory entries.
- e** Encrypt the zip file with a password (password prompt before starting the zip archive)
- f** Freshen the zip file with modified files.

-F	Fix a damaged zip file (not guaranteed to work).
-g	Grow a zip file, add more files to it.
-h	Show the help screen.
-i	Include only specified files. Example: <code>tar myfile.tar -i *.dat</code> will only include files with <code>.dat</code> ending.
-j	Just store files, do not save the directory structure.
-k	Attempt to make zip file MS-DOS compatible.
-L	Show the zip license
-m	Move files and directories to the zip file, deletes source files after they are compressed into a zip file.
-n suffix	Do not compress files with a certain type of suffix. For example, the command <code>zip myfile.zip -n .zip *</code> will not try to compress source <code>.zip</code> files into <code>myfile.zip</code> .
-o	Set the last modified time to the oldest modified time found among the source files.
-R	Save all files and directories under the source directory in a recursive manner.
-t mmddYYYY	Ignore file that are younger than the given month/day/year.
-tt mmddYYYY	Ignore files that are older than the given month/day/year.
-T	Test the integrity of the zip file, make sure it is good.
-u	Update zip file with the new and modified files (<code>-f</code> option only updates files, does not add new files).
-v	Verbose output, display additional text messages.
-x	Extracts files out of a zip archive.

The `unzip` utility supports the following options:

unzip The `unzip` command supports the following options:

`unzip <options> <filename.zip> <files to be decompressed>`

`<options>`

-l	Lists the files in an archive.
-o	Overwrite existing files without prompting.

- p** Extract files to STDOUT.
- t** Test the archive files.
- T** Set the timestamp on the archive file to the newest source file in the zip file.
- u** Updates existing files.
- v** Verbose output, displays verbose output during the extraction.
- z** Only displays the archive comments.

Solaris 9 also includes a zip/unzip utility called **gzip**. This program works much the same as zip/unzip, but it is not completely identical. For information on **gzip**, type the command **man gzip**.

The compress and uncompress utilities

The **compress** and **uncompress** utilities are different from other compression utilities. These utilities compress and decompress just a single file. These utilities are often used on files that are already compressed. For example, a system administrator could have a file named **myzipfile.zip** on a server. He or she would then type the command **compress myzipfile.zip** and create a file named **myzipfile.zip.Z**.

The **compress** and **uncompress** utilities use the Lempel-Ziv compression coding. The **compress** utility replaces the original **<filename>** with a new file with a **<filename.Z>** extension. The new file with the **.Z** file extension can then be copied or moved to another location or server. Once the **.Z** file arrives on the target system, the system administrator uses the **uncompress** command to extract the original source file.

The **compress** utility supports the following options:

compress < options > -b <9-16> <filename>

< options >

- c** Write to standard output. No **.Z** file is created.
- f** Force compression of a file. If **<filename.Z>** already exists, rewrite the file. If the **-f** option is not used and **<filename.Z>** exists, the user is asked if the file should be overwritten.
- v** Verbose output. Includes the percent of increase or decrease in a file's size when it is compressed.
- b (9-16)** Set the number of compression bits. A lower number of compression bits will result in a larger file that is faster to create. A higher number creates a small file that takes more time to create. The values 9 - 16 are the smallest and largest values allowed.

<filename.Z> Name of the compressed file.

The zcat utility

The **zcat** utility is used to extract **.Z** files directly into STDOUT output. This is useful when a system administrator comes across a file with the extension **.tar.Z**.

The one-line command

```
zcat myfile.tar.Z | tar -xvf -
```

can be used to uncompress a `.tar.Z` file to STDOUT.

The `dd` Command

The `dd` command is most commonly used to copy the full contents of one hard drive to another hard drive. This command is unique in that it can be used to copy files with different data formats, such as different block sizes and record lengths. Because of this unique characteristic, copying hard drives to tape and then back to a different medium is possible.

The `dd` command supports the following format:

```
dd if=<input file or raw_device_name > of=< output file or raw_device_name >
```

Lesson 18.4 Using the `dd` Command

In this lesson, you will use the `dd` command to copy the contents of a slice onto a floppy drive. A standard 3 ¼" floppy diskette can only hold 1.44 MB of data., so make sure to pick a slice that has less than 1 MB of data on it. After the slice is copied to a diskette, the `newfs` command is used to create a new file system on the old slice. The `dd` command is then used to restore the data back to the slice again.

WARNING

In this lesson, a floppy diskette and a hard disk slice are used to demonstrate the `dd` command. All existing data on the diskette and the hard disk slice will be erased (although the data on the slice will be restored at the end of the lesson).

Before starting this lesson, back up any critical information on the diskette that you will use, and on the hard disk slice.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command

```
dd if=/dev/<raw_device_name_of_a_slice> of=/vol/dev/aliases/floppy0
```

For example, the command

```
dd if=/dev/rdisk/c0t0d0s6 of=/vol/dev/aliases/floppy0
```

will copy slice 6 of the first hard drive to a diskette.
This command copies data from the input file (`if`) to the output file (`of`). The input file is the hard disk slice that you have chosen, and the output file is the floppy diskette. Use the raw device name of a slice that has some data, but less than 1 MB total. A slice name is something like `/dev/rdisk/c0t0d0s6..` If you need to review slice naming conventions, see Chapter 13.
4. Type the command `newfs /dev/rdisk/<raw_device_name_of_a_slice>`
This creates a new file system on the slice.
5. Type the command

```
dd if=/vol/dev/aliases/floppy0 of=/dev/rdisk/<raw_device_name_of_a_slice>
```

This command copies data from the input file (the diskette) to the output file (the hard disk slice).

The **dd** command can also be used to copy the contents of one hard drive to another hard drive. The drives do not need to have the same size data blocks, and they can have different file structures. However, the target drive (the one *to* which you are copying) must be equal to or larger in size than the source drive (the one *from* which you are copying).

Be aware that using **dd** for this operation takes a very long time. In the past the author has used **dd** to copy one 14 GB drive to another. It took all night! It's quicker to make a mirror copy of a disk with the volume management tools, synchronize the disks, then remove the volume management tools. Creating a mirror disk and then removing the mirror disk takes about one hour.

Lesson 18.5 Using **dd** to Copy a Hard Drive

In this lesson a workstation is brought down to the **OK** prompt in a graceful manner. The Solaris 9 Installation CD is used to boot to a CDROM image of the Solaris 9 operating system. The command **dd** is used to copy one hard drive's contents onto another hard drive.

Remember that the target drive (*to* which you are copying) must be equal to or larger in size than the source drive (*from* which you are copying). This lesson assumes that you are copying your *system drive* to your *second hard drive*

As mentioned in the warning below, make absolutely sure you know which hard drive is the *system disk* and which hard drive is *second hard drive*. In most cases, the system disk is **c0t0d0**. Double check this information!

WARNING

- This lesson shows how to use the **dd** command to copy one hard drive to another hard drive. It carries a higher than usual risk of damaging vital system information. There other methods of doing the same operation (which are not covered in this book). If you don't think you will ever need to copy one hard drive to another, consider skipping this lesson.
- Before doing this lesson, make sure you know what drive the Solaris 9 operating system is installed on!!! If you make a mistake, you will have to re-install Solaris 9 again.
- As a suggestion, type the command **mount** and notice the Logical Device Name (usually **/dev/dsk/c0t0d0s0**) of the disk that holds the root file system. Do not copy over this disk, or the Solaris 9 operating system will be destroyed!
- This lesson requires that you bring down the server, and the commands used in this lesson can take many hours to complete. Do not start this lesson unless you can afford to have your Solaris 9 system unavailable for several hours. The best time to start this lesson is just before you retire for the night.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command **init 0**

*If the system is not powered on, power on the system and press the STOP + A keys just before the operating system starts to load. Bring the system down to the **OK** prompt in a graceful manner.*

4. Insert the Solaris 9 Install CDROM.
5. Type the command **boot cdrom -s**
At this point in time a CDROM image of the Solaris 9 operating system is being set up. The CDROM image of Solaris 9 is useful for disk maintenance because the hard drive are sitting dead. This is good for disk maintenance because some commands can not be used on a live mounted slice.
6. *This step assumes that you are copying slice 2 of the system drive (remember that slice2 is a symbolic slice that represents the entire drive) to slice 2 of the second hard drive. If either the source or target drive is different than shown here, change the command in this step accordingly.*
Type the command **dd if=/dev/rdisk/c0t0d0s2 of=/dev/rdisk/c0t2d0s2**
This command could take all night to complete. Count on about 1 hour for every gigabyte of disk space.
7. Type the command **init 0**
8. Type the command **boot**
9. Open a terminal window.
10. Type the command **mkdir /2nddrive**
11. Type the command **mount /dev/dsk/c0t2d0s0 /2nddrive**
12. Type the command **cd /2nddrive**
13. Type the command **ls**
This command shows the contents of the second hard drive (to which the copy was made). The directories should look like the directories found under the root (/) file system of the system drive (from which the copy was made).
14. Type the command **cd /**
15. Type the command **umount /2nddrive**

Key Points to Remember

Knowing how to back up and restore a server is vital, particularly when a server crashes. It is important to understand how to use the **ufsdump** and **ufsrestore** commands to bring the server back to life. Utilities to compress/uncompress files, such as **tar** and **zip**, are very useful for saving files and directories into a single easy-to-transport file. Just remember that not all compress/uncompress utilities are compatible, especially across platforms. The **dd** command can be used for copying one hard drive to another, although it is not the ideal utility for this job.

Chapter 19 The Solaris Management Console

Lessons in This Chapter

The entire chapter is basically one big lesson

Introduction

The purpose of this chapter is to show some of the basic navigation features of the SMC (Solaris Management Console). This chapter also shows how to use some of the applications in the SMC. Other SMC-based applications will be covered in the appropriate subject chapter. For example, the Enhanced Storage Tool 1.1, also known as Sun Volume Management, is covered in Chapter 26.

The Solaris Management Console (SMC) can be thought of as a GUI toolbox for several different types of applications. These applications provide a wide variety of administration functions, including creating disk volumes, checking on processes and providing system information. The SMC can be used on the installed host system or on foreign Solaris systems that also have the SMC installed.

Note: Sun also has a product with the same initials, called "Sun Management Center." This chapter describes only the Solaris Management Console.

Starting the SMC

To start the SMC from a Terminal window, type the command `smc &` as the root user. Non-root users can also use SMC, but with restrictions on what tools and displays are available. Do not close the Terminal window in which the `smc &` command was typed, or the SMC session will also die.

Another way to start the SMC is to right click anywhere in unoccupied desktop space to display the Workspace menu. Once the Workspace menu appears, left click on Tools menu choice, then left click on the Solaris Management Console Icon.

Figure 19.1 is a screen shot of the SMC Icon in the Workspace menu. The SMC icon can also be seen in almost every popup menu and administration window inside the CDE.

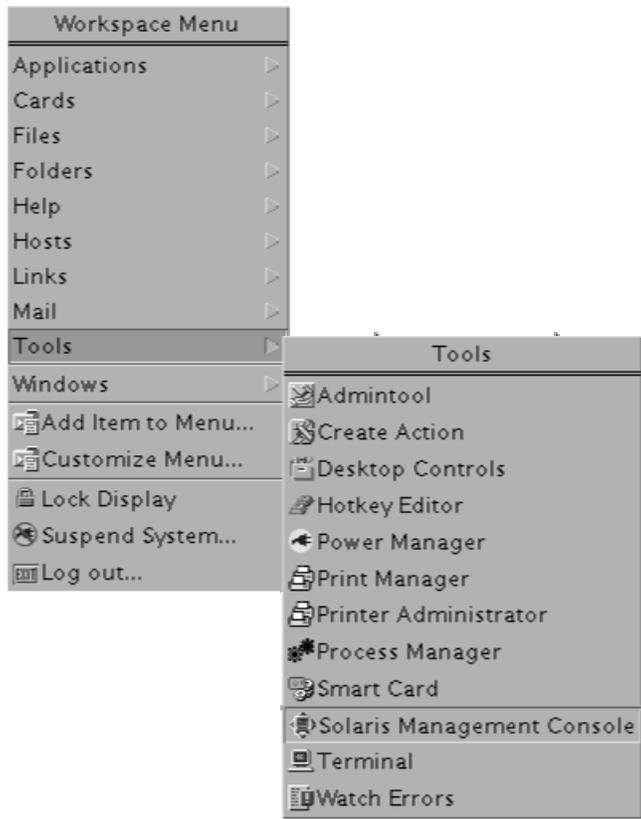


Figure 19.1 The SMC Icon in the Workspace Menu

After a brief splash screen is displayed, the SMC starts. Each time it starts, it takes a little bit of time to examine the operating system. Figure 19.2 is a screen shot of the Solaris Management Console window.

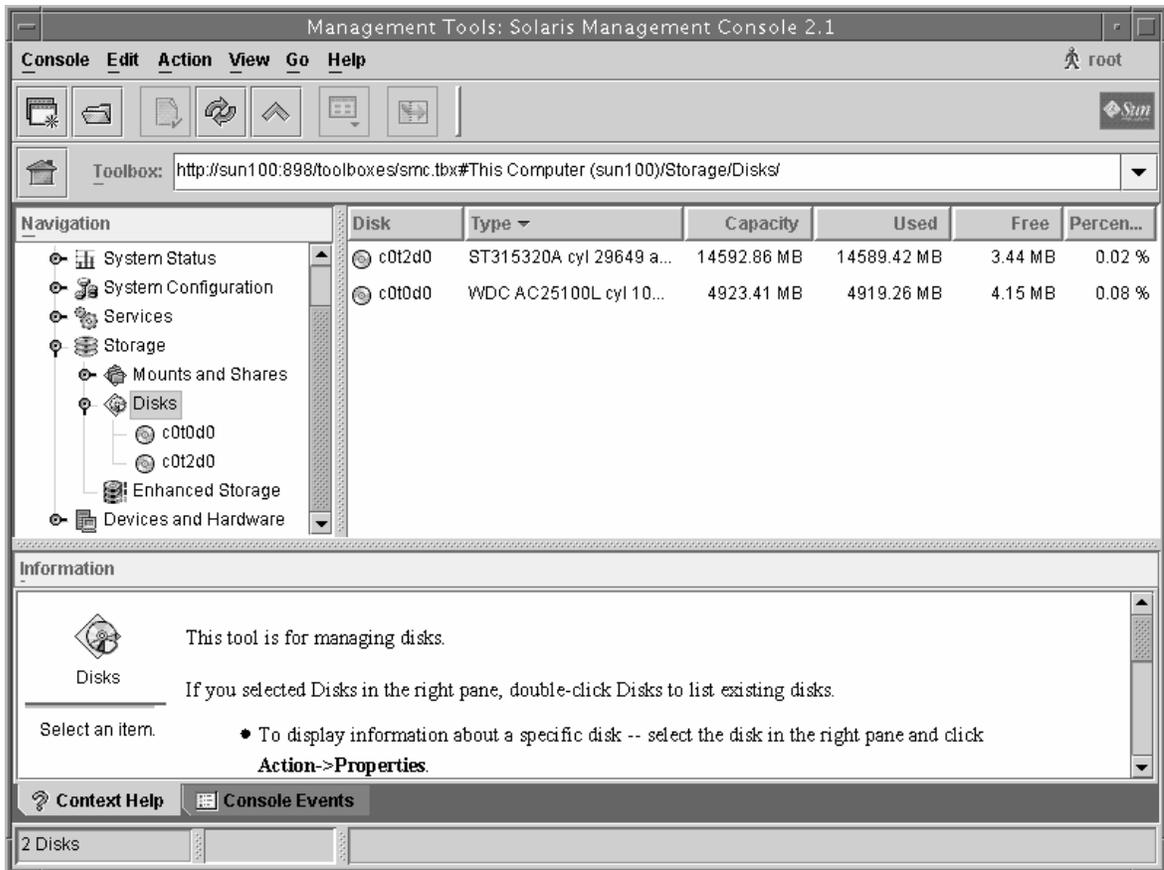


Figure 19.2 Solaris Management Console Window

SMC Components

There are three main software components that make up the SMC. They are:

The Solaris Management Console Server This server contains the base software.

The Solaris Management Console Client Also called “The Console.” This can be used on almost any client that supports Java.

The Solaris Management Console Editor A GUI that is used to edit the SMC toolbox. Basically, this lets an administrator create a custom SMC desktop with the tools most commonly used.

The SMC acts as a GUI toolbox for the following programs:

Solaris System Information 1.0 This tool provides two types of information. The first type is *static system information*, such as hostname, IP address, number of CPUs, and amount of memory. The second type dynamic system information, such as virtual memory used and CPU usage.

Solaris Log Viewer 4.1 This application shows the various log files created by the **syslog** process and system messages.

Solaris Process Manager 1.1 This tool lets the system administrator view processes, kill bad processes and perform basic process management functions (a process can be thought of as a program in UNIX). System administrators can use this tool on all system processes. Ordinary users can use it only on process that they themselves created.

Solaris Performance Manger 1.0 This application is really nice. It shows system information (such as memory used and CPU time) broken down by user or project.

Solaris User Manager 4.1 This tool performs the same types of functions that the Admintool did in the past. The only difference is that this user manager has a lot of nice wizards and templates that make it easy to work with large numbers of users at the same time.

Solaris Project Database Manager 1.0

Solaris 9 uses what are known as *projects*. A project is a collection of users and applications that can be monitored and restricted as a whole. This GUI tool provides good information on what projects exist, as well as some limited project management capability.

Solaris Computers and Networks Manager 4.0

This tool is used with IP addresses on networks and other Sun workstations. The nice thing about this tool is that it gives a bird's-eye view of the network and other Sun systems.

Solaris Patch Manager 1.0 This application shows the patches on the system and lets an administrator add and remove patches with a really nice GUI interface. Understand that the Admintool is going to be discontinued in the future, so this will be one of the only remaining GUI patch tools on Solaris 9.

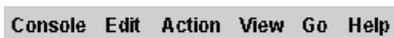
Solaris Job Scheduler 1.1 This is basically a front end GUI for the **crontab** command. Jobs (such as "run a program at a specific time and date") can be created and modified by this tool.

Enhanced Storage 1.1 This provides what is known as Volume Management. A Volume is a collection of hard drives under a RAID 0, RAID1 or RAID 5 image. Chapter 26 discusses this tool in detail.

Solaris Serial Port Manager 4.1 This tool changes the settings on the serial or parallel ports attached to a SPARC-based system. This is a wonderful tool to use when setting up a modem or terminal on a SPARC-based system.

Using the SMC

On the top of the SMC screen is the Menu bar. Figure 19.3 shows the SMC Menu bar.



Console Edit Action View Go Help

Figure 19.3 The SMC Menu Bar

The Menu bar has the following items:

Console

- New Console** Start a new Console window. This is just like starting multiple copies of Microsoft Word or multiple copies of Internet Explorer.
- Open Toolbox** Open a toolbox according to the user's customized settings.
- Preferences** Customize SMC by changing preferences such as font size, whether tools should be loaded only when used or all loaded on start, and authentication of SMC users.
- Console Events** When a user does something in SMC, the SMC records a log of the activity. The Console Events menu item records these events. The Console Events log is a useful resource for system security audits.
- Close** Close the SMC window. If there are multiple SMC windows, this command will just close the last one opened. If this is the last SMC window, the SMC session will come to an end.
- Exit** Close all SMC windows and end the SMC session.

Edit

- Select All** Select all the items in the View pane (the pane on the right).
- Properties** If a selected tool has properties, this menu selection shows them.

Action

If a selected tool has various actions, such as Open, Create Solaris Partitions..., and Copy Disk Layout... this menu dynamically changes according to the selected tool. The choice **Copy Disk** does not exist for the System Status tool.

View

- Show** Select which panes to show. The Tool bar, Location bar, Status bar, and the Navigation and Information panes can also be shown or hidden.
- View As** Choose whether to view information Web Style or in Panes,. Also controls font sizes.
- Columns...** Choose which columns are displayed in the View.
- Sort by** Sort by whatever columns are visible in the View pane. Left click on a column to sort the displayed items by the column heading.
- Show Commands**
Show the commands associated with a tool during the current session.
- Filter** Filter dialog boxes for what is shown in the View pane.
- Refresh** Refresh the contents of the SMC.

Go

- Up Level** Move up one level in the tool hierarchy.

Home Toolbox Return to the original SMC toolbox that started the session (depending on the preferences selected).

Help A typical help menu with Overview, Contents, and Index.

The Tool Bar

The Tool bar is directly under the Menu bar. Figure 19.4 shows the Tool bar items.



Figure 19.4 SMC Tool Bar Icons

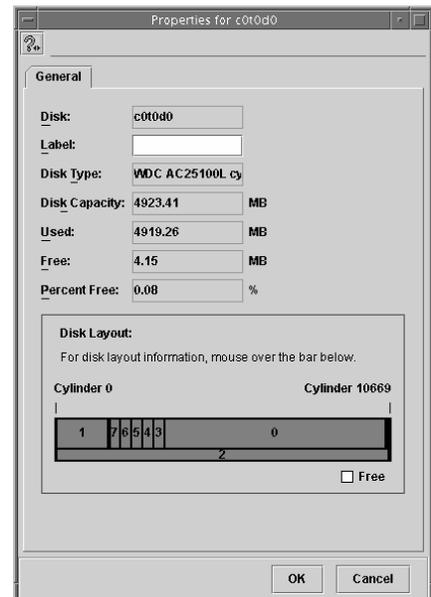
The Tool bar gives the user small icons to click on to create new views, copy, paste and perform other simple navigation actions. This is just like using the main toolbar in Microsoft Word for the same types of operations. The Tool bar changes dynamically, depending on what SMC tool is being used.

The SMC Tool bar icons, from left to right, are:

New Console When this button is pushed, SMC generates a new SMC console window that looks just like the original SMC console window.

Open Toolbox Opens a new toolbox, either on another server or on the host server.

Properties Pops up a Property window for a device or item. Some Property windows only show information about an item, while others let users change give detailed information on a device. For example, the Property window shown here shows the properties for a hard drive. This hard drive property window lets the SMC user change the disk's label. This screen shows detailed information about the hard drive.



Refresh Refresh the View pane.

Up Level Moves the selection in the Navigation pane up in the navigation tree. Just for reference, the system is the top part of the tree, while a device like the serial port would be the bottom of the tree.

View As Lets the user choose the type of view to use (Large, Small, List, Details, or Web Style) in the View pane. The Details selection is the best way to a new user to view the icons. Although fewer icons are displayed on the screen with this choice, it does provide a short text description of the icon to the left of the icon.

Filter Lets the user filter the different column headings in the View pane. The user can select a console object (such as Disks, Ports, or Log View) and perform a keyword search on the name or description of the item. Once the filter criteria are set, left click on the Filter button to start the filtering. The Navigation pane and View pane will have the phrase "FILTER VIEW" at the bottom of the background. To undo a filter, click on the Filter button again to open the Filter

window, and then click on the Unfiltered button. Once an item is unfiltered, the original white background returns.

Dynamic Buttons These buttons are automatically added or removed from the Tool bar, depending on what item is selected in the Navigation pane. For example, when the Solaris Processes Manager 1.1 is selected, buttons labeled Suspend and [] Refresh Every [seconds] are shown. These buttons are not shown when the Enhanced Storage Tool is used.

The Location Bar

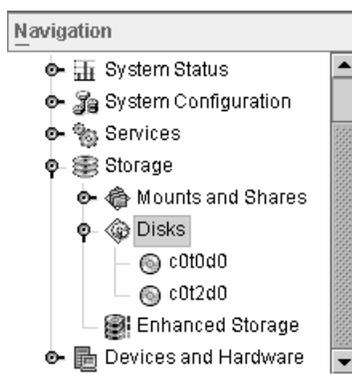
The Location bar is directly below the Tool bar. Figure 19.5 shows the Location Bar.



Figure 19.5 The SMC Location Bar

The Location bar has a small icon (that looks like a house) on the left side. When this icon is clicked, the user is returned to the initial Navigation view that appears when the SMC first starts up. The text box to the right of the Home Toolbox icon works just like the text box in most Web browsers. You can enter a URL here, or you can return to previous locations by clicking on the drop down arrow at the right of the text box.

The Navigation Pane



The Navigation pane is on the left of the Console window. This pane shows the components of the computer in a tree view. Figure 19.6 shows the Navigation pane. The branches of the tree can be enlarged or collapsed, depending on what items are of interest to the system administrator. This tool can show the devices on one computer, or on several computers that are linked to the same SMC.

Figure 19.6 The SMC Navigation Pane

The Navigation pane holds the following applications:

- | | |
|-----------------------------|--|
| System Status | Shows information about the selected host. |
| System Configuration | Shows Users, Projects, Patches and All the computers and networks found by the host. |
| Services | Contains the Scheduled Jobs tool. This tool lets a user schedule a program to run at a given time, just like the crontab command. |
| Storage | Shows all hard drives and tape drives on the system. Clicking the Enhanced Storage icon starts up the Sun Volume Manager. |
| Devices and Hardware | Shows specific types of hardware that can be controlled by SMC. In this case, the system's ports are the only such devices. |

On the left of every navigation item is a small blue icon that looks like a magnifying glass. This icon is called the Turner icon. When the icon is left clicked it expands or collapses the tool hierarchy. You can also expand or collapse the tool hierarchy by double clicking on a tool.

The View Pane

Figure 19.7 shows the Details view of the View pane. This pane shows the objects in the computer or shows sub-menu choices for items in the Navigation pane (the pane on the left of the Console window.) Figure 19.7 shows the two hard drives that are installed on the author's SunBlade 100 workstation.

Disk	Type ▾	Capacity	Used	Free	Percen...
c0t2d0	ST315320A cyl 29649 a...	14592.86 MB	14589.42 MB	3.44 MB	0.02 %
c0t0d0	WDC AC25100L cyl 10...	4923.41 MB	4919.26 MB	4.15 MB	0.08 %

Figure 19.7 SMC View pane

On the top of every column is a Column button. When these buttons are right or left clicked they sort the information by the Column button's topic. In Figure 19.7, the disks are arranged in alphabetical order by the Type column. If the Capacity button is clicked (by either mouse button) the information will be sorted by the capacity of the disks.

Each time you click a Column button, the data will be re-sorted by that column. Notice the down-pointing triangle in the Type column in Figure 19.7. This means the data is being sorted in ascending order by that column. When the triangle points down, the data is being sorted in descending order.

The Help Pane

Figure 19.8 shows the Help pane.

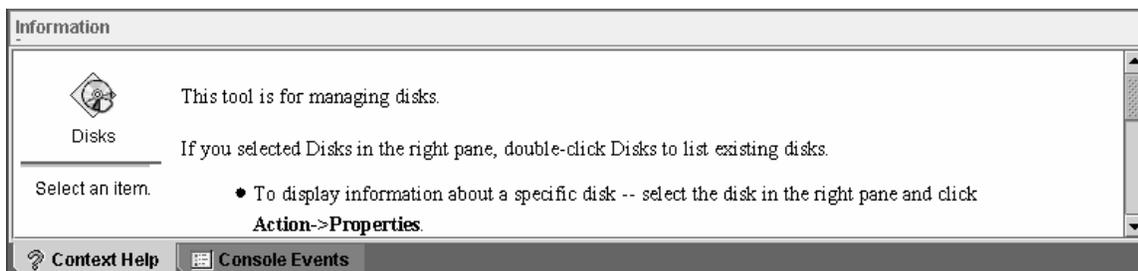


Figure 19.8 The SMC Help pane

The Help Pane is the pane on the bottom of the SMC. This screen gives useful information about the item highlighted in either the Navigation pane or the View pane. This is an excellent innovation by Sun Microsystems that makes the SMC very easy to use.

Using the SMC

This section uses the System Status tools to demonstrate how to use the SMC. Most of the other SMC applications are shown in other chapters in this book.

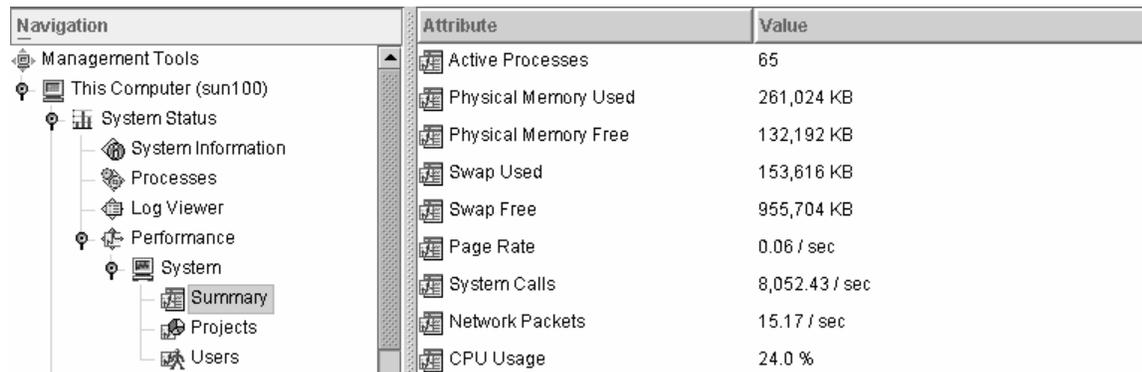
The SMC is a very easy to use tool. Even novice users can use the SMC after only a couple of hours of *playing around* with the application. Keep in mind that the screen shots in this book may look different than what you see. This is because there are different versions of SMC, and because the SMC GUI is a dynamic tool that adds or removes items, depending on the hardware found on the system.

Solaris System Status

The System Status folder will be used to demonstrate how SMC works. To open this folder, click on **System Status** in the Navigation pane, as shown in Figure 19.6. This application has four utilities, as shown in the following table:

Program	Icon to Click to Run Program	Description of Program
Solaris System Information 1.0	System Information	Shows static and dynamic system information.
Log Viewer 4.1	Log Viewer	Shows various syslog files and program log files.
Solaris Process Manager 1.1	Processes	Lets users perform system administration tasks on the operating system's processes.
Solaris Performance Manager 1.0	Performance	Shows system resources broken down by user or project

Figure 19.9 shows an expanded view of the System Status Folder and its applications, in the Navigation pane.



Navigation	Attribute	Value
Management Tools	Active Processes	65
This Computer (sun100)	Physical Memory Used	261,024 KB
System Status	Physical Memory Free	132,192 KB
System Information	Swap Used	153,616 KB
Processes	Swap Free	955,704 KB
Log Viewer	Page Rate	0.06 / sec
Performance	System Calls	8,052.43 / sec
System	Network Packets	15.17 / sec
Summary	CPU Usage	24.0 %
Projects		
Users		

Figure 19.9 The System Status Folder in the Navigation Pane

Solaris System Information 1.0

This application gives some very generic overall system information. Most of the information is static, such as the number of CPUs and number of disks. Figure 19.10 shows the System Information 1.0 application.

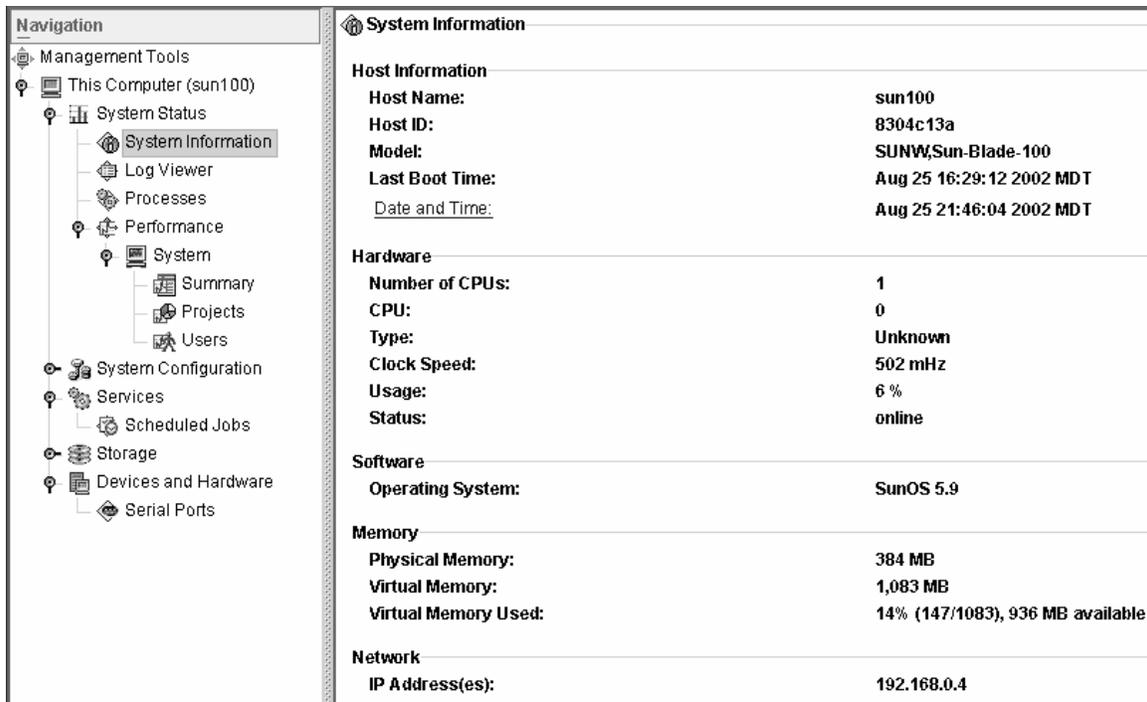


Figure 19.10 Solaris System Information 1.0

This is a convenient tool to use for documenting your server's configuration. There are some key items to focus in on:

- **Host name:** This is the network name that is used with commands like `ping server32` or `telnet server32`. The author's workstation has the hostname `sun100`.
- **Last Boot Time:** Shows when the server was last rebooted. This is a key piece of information to have if a user complains that the server was inaccessible. This shows if the server was still running at the time. The system's date and time can be reset here by clicking on **Date and Time**. The day, month, year, hour and minute can easily be changed from this application.
- **Hardware:** If a server seems very slow, look at the "Status:" line in the Hardware section. If a CPU is offline, that could actually mean that the CPU is down. The "Usage:" line in the Hardware section should be 80% or less. If this statistic is in the range 90%-100%, fewer applications should be run simultaneously on the server, or additional CPUs need to be added to the server. Remember, though, that with Solaris 9 licensing, each CPU requires a separate license fee from Sun.

Solaris Process Manager 1.1

The Process Manager shows the system processes (a process is a program) that are running on the server. Figure 19.11 is a screen shot of the Process Manager.

Process Name	PID	UID	PPID	PR	NI	VSZ	SSZ	State	Priority	Session	Working Set
init	1	root	0	0.0	1208			Sleeping	system	1	/etc/finit-
pageout	2	root	0	0.0	0			Sleeping	system	0	pageout
fsflush	3	root	0	0.1	0			Sleeping	system	0	fsflush
sac	394	root	1	0.0	1784			Sleeping	system	1	/usr/lib/se

Figure 19.11 The Process Manager

The processes are shown on the right. If a process is right clicked, a popup menu appears, as shown in Figure 19.12.

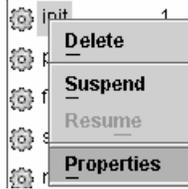


Figure 19.12 Process Right Click Popup Menu

This popup window lets the system administrator Delete (kill) or Suspend (temporarily halt) a process. Clicking on Properties brings up a popup menu, in which the properties of a process can be viewed. Figure 19.13 is a screen shot of the Properties window of the `init` process.

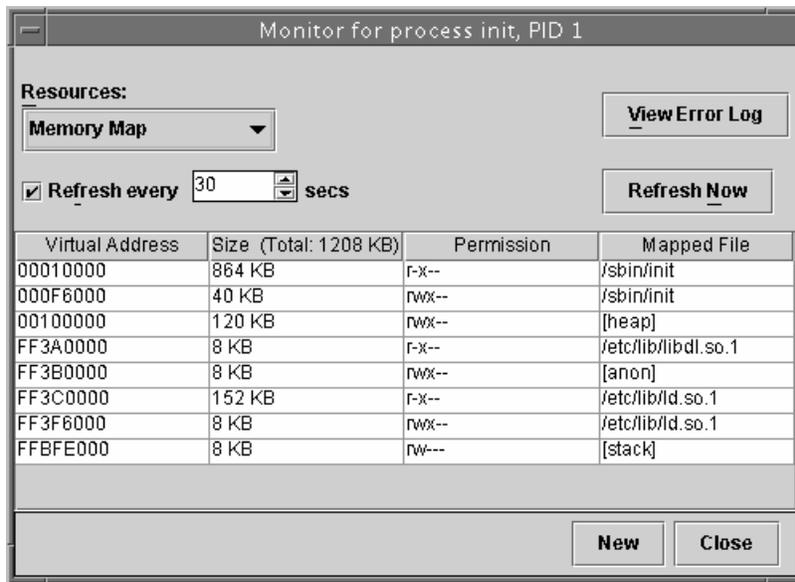


Figure 19.13 Properties of the `init` Process

Take a look at the first process shown in Figure 19.13. The nice thing about this window is that it shows what command started the process (in this case `/sbin/init`). If a server has a runaway process (a process that tries to grab all the system memory and CPU time), the Solaris Process Manager shows what program the process was called by. After killing the runaway process, you may want to shut down and examine the program that called it. That program may need a software patch, a new configuration, or a workaround.

At the upper right of the window in Figure 19.13 is a button labeled “View Error Log” To see the error log for a given process, highlight the process by clicking on its line, then click the View Error Log button.

Note: The properties of some processes can not be viewed, because those processes are so critical that analyzing one of them might halt it. The page out process is one such process.

Quick Tip

- Sometimes when a process is right clicked, the error message **Error log for page out, PID, Memory Map** is displayed, along with the warning message

Unable to obtain this type of information

Reason: obtaining information for processes may require that the process be stopped and certain critical system processes necessary for system reliability must not be stopped.

As the message indicates, there are some processes that need to be stopped and restarted to gather the processes information. These are usually the system processes (PID < 100).

- This message will be saved in the error log for that process, but the process itself will not be harmed by the attempt to view it.

Log Viewer 4.1

The Log Viewer shows the various log files associated with the SMC and the system. Figure 19.14 shows some generic log information in the Log Viewer.

System Status	Log	8/20/02 11:56:22 AM	sun100	root	Solaris_Patch	Application log	Patch add
System Information	Log	8/25/02 5:38:37 PM	localhost	N/A	Solaris per CP...	Application log	Perfrmon P
Log Viewer	Log	8/25/02 6:00:35 PM	localhost	N/A	Solaris per CP...	Application log	Failure
Processes	Log	8/25/02 5:54:58 PM	localhost	N/A	Solaris per CP...	Application log	Failure
Performance	Log	8/25/02 5:38:37 PM	localhost	N/A	Solaris per CP...	Application log	Failure
System	Log	8/25/02 9:46:03 PM	sun100	root	TimeZone	Application log	TimeZone
Summary							
Projects							

Figure 19.14 Log Viewer 4.1 Display

The `syslog` file and the `/var/adm/messages` file can be opened through the Log Viewer. To open a log entry, right click on it. A popup menu will display, as shown in Figure 19.15. (This is the same menu that appears if you left click on the Action menu in the Menu bar at the top of the screen.) You can now open the related log files and view all the log file entries that relate to the entry on which you right clicked, or to any other log entry. Figure 19.15 shows this popup menu.

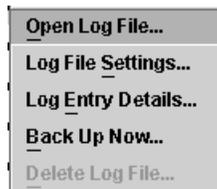


Figure 19.15 The Log Entry Popup Menu

Some of these log files only relate to the operation of the Solaris Management Console Application. These log files are saved in the `/var/sadm/wbem/log` directory. These log events are created by the WBEM logging

service. A brief description of each log event is shown at the end of the line for that event. The log events can be arranged by right or left clicking on the Column buttons on the top of the view menu.

The options in the Log Entry Popup menu perform the following tasks:

- Open Log File... Lets the user open the `/var/log/syslog` and `/var/adm/messages` files. A file with a name like `Log.06/12/2002.05:24:48` is the SMC log file. Any of these files can be opened in read-only mode.

- Log File Settings... Lets the user specify the directory where the SMC log file should be saved. The root user can also specify how large the log file should become before it is closed out (and saved) and a new log file is created. The number of SMC backup log files can also be set here. If **Enable System Logging** is checked, messages are written that to the `/var/log/syslog` file from the `syslogd` daemon.

- Log Entry Details... When a log file is highlighted, gives a detailed description of that log file. The up and down arrows allow the user to track forward and backward through the messages.

- Backup Log Files... When this is selected, the current log file is backed up and a new log file is created. Understand that if there is a set number of log files, the oldest log file will be erased. When a new log file is created, it will be empty. Entries from the log file that was just backed up will not appear in the View pane.

If for some reason a log entry does not appear in the View pane, left click on View and then on Refresh. There might be a short delay between events and when they are logged in the log file. Also, network communications or a heavily loaded server might slow down the logging of events. Be patient; the logged event will eventually show up.

The Solaris Performance Manager 1.0

The Solaris Performance Manager is the next application the System Status Folder. Figure 19.16 is a screen shot of the Solaris Performance Manager, showing system summary information.

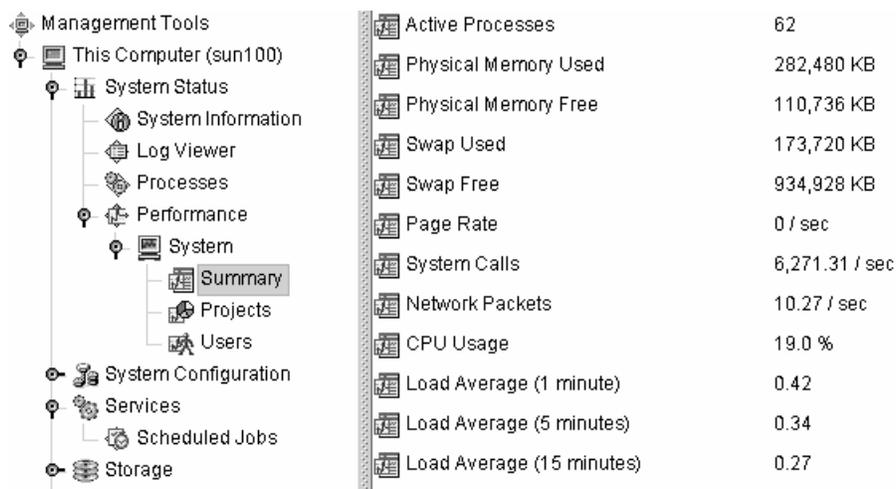


Figure 19.16 System Summary Information in the Performance Manager

The System Summary information is different that the System Information reported by the Solaris System Information 1.0 application, which only reports on a very limited range of hardware. As shown in Figure 19.16, the system summary screen shows many real time server statistics. These include real time performance statistics regarding the processor, memory usage, and system loads.

You can choose how the System Summary information is displayed. To do this, choose Action, Performance Bar Preferences on the Menu bar. Figure 19.17 shows the Performance Preferences window.

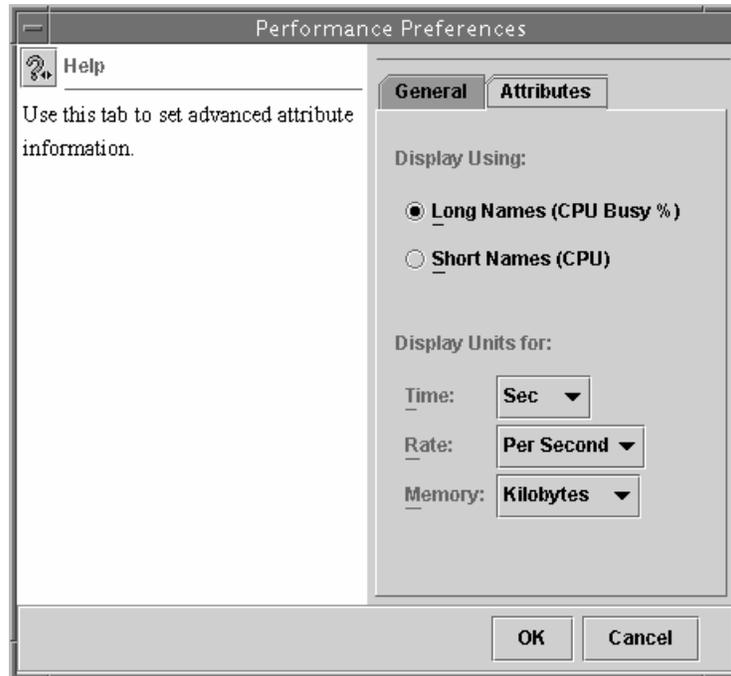


Figure 19.17 The Performance Preferences Window

In this window, you can select the refresh rate (under the General tab). The Attributes tab lets you specify the following preferences:

- Type of names displayed (short names or long names)
- Display units for time (Seconds, Milliseconds, Microseconds, or Nanoseconds)
- Rate of display (Per Second, Per Minute, or Per Hour)
- Units for display of memory amounts (Kilobytes, Megabytes, or Gigabytes).

One annoying characteristic of the Performance Manager is that by default, the View pane is refreshed every 30 seconds. It takes the program about twenty seconds to refresh the screen. During the refresh time, the screen turns blank. The user can only look at the statistics for about five or ten seconds during long periods of blank screens. How annoying !

The next category of System Resource statistics is Projects. A Project can be thought of as a collection of programs from something like HP Openview, the Apache Web server, or an Oracle 9i server. The Projects view answers questions like “How much of the server's resources is Oracle 9i currently using?” Figure 19.18 shows a screen shot of the Projects view.

User Name	Num Processes	% CPU Time	% Memory Used	Resident Set Size ...	Process Image
system	30	0.02	13.15	48,888	7
user.root	27	18.14	52.85	195,984	29

Figure 19.18 The Projects View

The Users icon lets you view the currently logged on users, and the resources that each user is using. Figure 19.19 shows this view.

User Name	Num Processes	% CPU Time	% Memory Used	Resident Set Size ...	Process Image
daemon	1	0.00	0.47	1,760	;
root	56	22.53	65.74	243,904	37

Figure 19.19 Users View, Showing the Resources that Users are Taking.

Understand that some user accounts are internal Solaris accounts. For example, the **daemon** and **root** accounts are internal system accounts that the operating system uses for its own purposes. These are not accounts that a user can log in with. Figure 19.19 shows these accounts, at a time when no regular users are logged onto the system. Any account in the `/etc/passwd` file with a UID (User ID) of less than 100 is an internal system account (except the root user). There are also some applications, like Oracle that require the creation of a user account that is used by the software program.

System Configuration

Near the bottom of the Navigation pane in Figure 19.16 is the System Configuration icon. This folder holds various utilities that can be used to configure a Solaris server. Some of these applications are covered below.

WARNING

Before a discussion of the Project Database Manager can start, it is critical that the reader understand the `/etc/project` file that forms the basis of the Project Database Manager.

Be very careful editing this file! Any mistake can cause the **get project** and other program entries not to assign projects properly, or may cause a project to crash.

Solaris Project Database Manager 1.0

A Project is used to manage and track the consumption of computer resources. The file `/etc/project` is a database text file for the projects that are currently on the system. One line per project is the rule! Each project is defined by one line; that line can not wrap to a second line. Understand that the Project Manager is a GUI front end for `getprojnet`, `getprojidv`, `vfgetprojnet` and other programming functions related to projects.

The `/etc/project` file contains the initial projects that were created when Solaris 9 was first installed on a system. These projects are:

```
system:0:::  
user.root:1:::  
noproject:2:::  
default:3:::  
group.staff:10:::
```

A system administrator can create additional projects. These projects will then appear in the `/etc/project` file.

Each line in `/etc/project` is organized by the following columns, separated by colons (`:`)

ProjectName:PROJID:Comment:Userlist:Grouplist:Attributes

ProjectName The project name. This must start with a letter. Other characters can be a letter, a number, or the underscore (`_`) character. Projects with a period (`.`) in the name are special system projects.

PROJID The project ID. This is a numerical identification of the project, similar to a social security number for a person.

Comment A text comment on the project. This makes the information more understandable.

Userlist A comma separated list of users that can use this project. If this list is blank, no users can use the project. If a user's name has an exclamation mark (`!`) that user is excluded from the project. If an asterisk (`*`) is used, anyone can use the project. If the combination "asterisk-exclamation-mark" (`*!`) is used, nobody can use the project.

Grouplist A comma separated list of groups that can use this project. If this list is blank, no groups can use the project. This is **NOT** the same type of "group" as the ones in the `/etc/group` file (which allows a user to set group permissions for a file). In this case, the term "group" relates only to permissions for using a project that is listed in `/etc/project`.

If the grouplist is blank, no groups can use the project. If a group's name has an exclamation mark (`!`) that group is excluded from the project. If an asterisk (`*`) is used, any group can use the project. If the if the combination asterisk-exclamation-mark (`*!`) is used, nobody can use the project.

Attributes A list of **name=value** pairs separated by semicolons (`;`). The name is the name of the attribute, and the value is the value assigned to the name. The value can use upper and lower case letters, numbers, and any of the following characters: underscore (`_`), front slash (`/`), hyphen (`-`), plus (`+`) and period (`.`). These **name=value** pairs are the attributes that define the project to programs for resource control.

A project can use `/etc/project`, or the project can use other project information sources, such as the NIS maps `project.byname` and `project.bynumber` or an LDAP server. NIS (Network Information Server) is a client/server architecture that stores and distributes the information that is commonly found in `/etc/project`. For now, just understand that a NIS server can provide project information. LDAP is another type of information server that can also provide project information. It is similar to NIS but is more powerful.

Key Points to Remember

This chapter demonstrated some of the basics of the SMC. Without exaggerating, the SMC is a very easy to use tool with all of its help screens and step-by-step wizards. The best advice the author can give is to play with the SMC on the test workstation. The real power of the SMC is its ability to work on multiple systems or user accounts at the same time. Most of its functions can be performed by other GUI tools (like the Admintool) or by command line tools. But remember, the nice thing about the SMC is that the administrator does not need to log into different systems and type the same commands again and again.

Chapter 20 What's New in Solaris 9

Introduction

This chapter introduces readers to some of the newest features in Solaris 9. This is only a quick description of these features. For readers who have never worked with Solaris 9, it might be a good idea to skip this chapter.

Security Improvements

One problem with Solaris 8 was that some high end security measures were not built into it. For example is that SSH (Secure Shell) was not bundled with Solaris 8. System Administrators had to download SSH from <http://www.sunfreeware.com> (sunfreeware.com is an excellent webpage that has precompiled Solaris software, try it!). Solaris 9, on the other hand, has many third- party security tools built in to the operating system.

Solaris Secure Shell (SSH)

One of the greatest security holes with Solaris 8 occurred when a user established a **telnet** session or **FTP** session between two servers. The **telnet** session did not use any form of encryption within the network communications. Any Solaris server, or any other UNIX server, could use the **snoop** command to view the network traffic between a workstation and a server. The **snoop** command captures any network traffic that goes through the system's Ethernet card and then saves this information to a text file. Later, a malicious user could troll through the snoop file for usernames and passwords to use for hacking.

In Solaris 8, users generally have to use one of two types of security measures to prevent this scenario. The first is to use a switch instead of hub. A hub is a piece of network equipment that has several Ethernet ports (often 8 ports, 16 ports or 32 ports) that communicate with each other on a bus. A hub connects several Ethernet cables, just like a railroad roundhouse of the 1800s connected several train tracks. All workstations connected to a hub "see" all the Ethernet traffic in the hub. A hub works at the Network layer of the OSI model. Because all Ethernet traffic can be seen by anything connected to a hub, Ethernet traffic can be snooped out of a hub, even by even foreign workstations and servers.

A switch does the same job as a hub, but it is slightly more intelligent. Instead of using a central bus to dump all the network communications, a switch routes traffic exclusively between two communicating ports. For example, if there is a server on port 12 and a workstation on port 3, port 12 and port 3 *talk* to each other. If John Smith establishes a **telnet** session with a server, a malicious user on the same server as John Smith (with the root user's password) can still **snoop** the network traffic and troll for John Smith's username and password. But using a switch also means that ports not directly involved can not view a communication. So the **snoop** command on a workstation attached to a switch can only view its *own* traffic. A switch prevents a third-party workstation from snooping traffic between two other workstations or servers. This lessens the potential security problem (but does not completely eliminate it).

The second way to prevent snooping of network traffic in Solaris 8 is to use a login utility named **SSH** (Secure Shell). **SSH** allows for authentication and for secure encrypted communications between a client and a server. This includes secure X graphics connections. This encrypted session uses the PGP (Pretty Good Privacy) schemes to pass keys between the server and the client. Snooping this traffic would be completely useless to a hacker. Unfortunately with Solaris 8, **SSH** was not built into the operating system as a native function. It must be obtained from the Internet and installed as an unbundled software package. This can be a very difficult and time-consuming task when several hundred servers are present in a company. Sometimes junior-level system administrators make mistakes, which renders the server inaccessible until a more seasoned administrator can fix the problem.

To compound the problem, there some software developers use **telnet** and **FTP** within their scripts. This creates a security hole that can only be solved by blocking telnet and FTP sessions on the server. Just as an example, the Veritas Netbackup Server can automatically download the Veritas Netbackup Client to Solaris servers. Unfortunately, if the FTP and Telnet ports are blocked, this automatic download can not be performed.

Part of the new security improvements in Solaris 9 is that it now supports SSHv1 and SSHv2 as native functions. If the recipient of a telnet session does not have SSH installed, Solaris 9 users will still be able to connect via that telnet session.

Internet Key Exchange (IKE)

IKE helps with key management with IPsec or IP Security. TCP/IP communication features two types of protocols. TCP is a Transport layer protocol in the OSI stack. This protocol resides in IP packets. The IP protocol is a Network layer protocol. It is possible to encrypt the information in an IP packet. Unfortunately, with Solaris 8 it tended to be rather difficult to maintain the security keys necessary for this encryption. IPsec on Solaris 9 automates the modification and replacement of keys.

Improved LDAP (Lightweight Directory Access Protocol) Security

LDAP is used to pass information between an LDAP client and an LDAP server. For example, email systems are a prime candidate for an LDAP system. Under Solaris 8, if a user wants to look up another user's email address on a global email list, the user's workstation can contact an LDAP server on the network. LDAP can be thought of as a *company database* server. In the past, one of the biggest security problems with LDAP was that hackers could spoof a client workstation's identity (sending requests to an LDAP server under the name of a legitimate client) and ask for LDAP information. This information could then be used for further hacking.

Solaris 9 has a built in LDAP server that is based on the Netscape Directory Server (as of the publication date, this was version 5.1). This server includes SASL (Simple Authentication and Security Level) For more information about Solaris 9's new LDAP security features, please see Chapter 30.

128-bit Encryption

Solaris 8 only supported 64-bit encryption. Solaris 9 supports 128-bit encryption. The more bits used in encryption, the harder it is for a hacker to unencrypted data with the *brute force method*. This method uses a computer program to generate and try every conceivable combination until a valid security key is found.

With 128-bit encryption the problem becomes much harder. Because each bit represents a zero or a one, 2-bit encryption could have a maximum of 4 possible values (2 raised to the 2nd power = 4). Similarly, 4-bit encryption could have 16 possible values (2 raised to the 4th power = 16). Each time a bit is added, the number of possible values doubles. Encryption that uses 128 bits has over 340,282,366,920,938,463,463,374,607,431,770,000,000 possible values. Because of the tremendous size of this number, the brute force method of cracking Solaris 9 is not a possibility. This 128 bit encryption support for Solaris 9 can be used by software developers within a wide range of software packages.

Kerberos Key Distribution Center (KDC)

Kerberos is a type of security algorithm. Kerberos Version 5 security is supported on Solaris 9. One of the primary Kerberos security tools in Solaris 9 is the Kerberos KDC (Key Distribution Center.) This server is used to issues tickets to clients. This database can be replicated onto backup servers if the primary KDC server goes down. The KDC can receive administration through a GUI and a command line interface.

Improved Process Controls

Most companies expect a Sun server to have 99.99 % uptime. If the system crashes or needs a reboot, this will affect the uptime of the server. To prevent system crashes and the need for reboots, Solaris 9 has extensive process controls. If a process is consuming too much CPU time or is starting to go out of control, the system administrator must be able to notice and correct these problems. The Solaris Resource Manager has been greatly improved from Solaris 8.

Fixed-Priority Scheduling

FX scheduler sets policy scheduling for processes used by applications and users. These processes are fixed. For example, `pricnt1` and `dispadmin` are two utilities that control the Fixed-Priority Scheduling. The FX class is the same priority as the FSS, IA and TS classes.

Solaris Resource Manager

The Resource Manager in Solaris 9 has improved features over the Resource Manager in Solaris 8. The resource manager lets the system administrator set constraints on the system resources used by processes and tasks. (A task is a collection of processes that have related activities). The Resource Manager also supports resource pools that are used to separate system resources. This works even after a system reboot. The fair share scheduler is also included in Solaris 9 and is used for fair delegation of CPU resources. The Resource Manager is controlled through the SMC (Solaris Management Console.)

Network Compatibility:

Most large sized companies do not want to be tied down to only one vendor for hardware or software. Because of this business model, when a company sets up its I.T. department, it looks for different types of servers that can communicate and work with each other. Sun Microsystems has gone to great lengths to ensure that its operating system will be compatible with other types of operating systems and software code.

Linux Support

Linux is a free UNIX-compatible operating system that runs on Intel (and other) platforms. One of the key issues Sun Microsystems has faced when dealing with Linux is the issue of software source code compatibility and software library compatibility. No large scale corporation would purchase new Sun equipment if all the programs written for Linux have to be re-written and recompiled for the Solaris operating system.

Solaris 9 now supports some of the more popular Linux commands. This is done so that scripts and programs that call on native Linux commands will not have to be recompiled and re-engineered for Solaris 9. Even in Solaris 8, Sun already used some of the same OpenSource programs as Linux (such as the Apache Web Server and DNS server). But Linux and Solaris have different operating system structures, which has led to some incompatibilities.

In addition to adding some native Linux commands, Solaris 9 now has more freeware and Open Source library routines built into the operating environment. In the past, these libraries had to be downloaded off the Internet or installed from the Solaris 8 Companion CDROM. These include GLIB and GTK+ and other common Open Source library routines. Solaris 9 has these software programs and libraries built into the operating environment.

Sendmail 8.12

Sendmail 8.12 is included with Solaris 9. Sendmail 8.12 now supports LDAP directory services and IPv6 addresses. Sendmail now listens for communications on port 587. This version of Sendmail handles security features better than previous versions, which would let other companies bounce spam messages off of the email server.

Mobile Internet Protocol

Mobile Internet Protocol is now supported in Solaris 9, with dynamically created interfaces. These implementations are able to send advertisements through dynamically created interfaces. The Mobile Internet Protocol starts after the **mipagent** daemon starts. The Mobile Internet Protocol was technically supported by Solaris 8 6/00 release. The Solaris 9 Mobile Internet Protocol also lets system administrators set up reverse tunnels from the mobile node's address to the home agent. Private addresses can be assigned to remote users.

BIND 8.2.4

The DNS server is also known as a BIND server. Solaris 9 uses a version of BIND that has been upgraded to release 8.2.4. Solaris 8 used version 8.0.2 BIND. BIND 8.2.4 supports the **ndc** command used to start or stop the **in.named** process. The new BIND supports the 3RESOLV interface so that it can be used in multi-threaded applications. The **dnskeygen** command is also supported, which updates DNS keys.

Solaris PPP 4.0

PPP (Point to Point Protocol) is used for communications over modems and other serial devices. PPP is a carrier protocol for underlying protocols like TCP/IP and SPX. PPP supports synchronous and asynchronous communications. PPP 4.0 also supports CHAP (Challenge Handshake Authentication Protocol) and PAP (Password Authentication Protocol). The command line utility **asppp2pppd** is provided to migrate PPP from previous version to PPP 4.0.

Windows 2000 Support

Solaris 9 has two features that help it work with Microsoft Windows 2000. These features are Samba 2.2.2 and client password change capability for Windows 2000.

Samba 2.2.2 is a utility that lets Microsoft Windows and UNIX systems read and write files directly to each other's file systems. This is an OpenSource SMB (Server Messenger Block) and CIF protocol stack that lets Microsoft Windows exchange and use file systems and printers on Solaris systems.

Solaris 9 now lets users change their user passwords on a Microsoft Windows 2000 server. This feature also supports the changing of passwords on an MIT Kerbose server.

What Happened to the GNOME 2.0 Desktop?

One of the biggest news flashes about Solaris 9 was that Sun would move away from the CDE (Common Desktop Environment) and move over to the GNOME desktop. The GNOME desktop is an OpenSource desktop that is commonly found on the Linux operating system. The GNOME desktop has a taskbar that looks a lot like the Windows 95 taskbar.

Unfortunately, the integration of the GNOME desktop did not take place fast enough for the release of Solaris 9. Currently, Solaris 9 has CDE version 5.50 as the default desktop. However, Sun Microsystems is providing a GNOME 2.0 download for Solaris 9. This can be obtained at <http://www.sun.com/gnome>.

Sun is only distributing this desktop as an "Early Preview of the GNOME desktop" for evaluation purposes. Sun claims that this desktop will be fully compatible with the CDE desktop that Sun is trying to move away from. Consider using the GNOME desktop on a low-priority desktop when it first comes out, because it is a bit risky to try new things on a server or critical desktop system. Power users might want to try and add GNOME to the Solaris 9 operating system themselves instead of waiting for Sun to add the desktop.

Disk Management

An important feature of Solaris 9 is its ability to handle read/write CDROMs. Solaris 8 had no native support for CD-RW drives. Another very strong point about Solaris 9 is its volume management. It is fairly easy to create RAID 0, RAID 1 and RAID 5 volumes on Solaris 9. Most of the tools are found in the SMC (Solaris Management Console).

Ability to Burn CDROMs

The `cdwr` command can be used with CDROM burners to create custom CDROMs with Solaris 9. Before `cdwr`, most Solaris systems had to use unsupported open source CDROM burning software. For information on `cdwr`, see http://www.sun.com/io_technologies/pci/removable.html.

Solaris Volume Manager

SVM (Solaris Volume Manager) is used to create RAID volumes. A RAID volume is a virtual partition that lies over several disks. This is transparent to users and applications. SVM supports RAID 0 (disk striping), RAID 1 (mirror a disk) and RAID 5 (disk striping with parity information). SVM is the next generation replacement to the Disk Suite Application used with Solaris 8.0. Solaris Volume Manager now supports Volume transaction logging devices. Transaction logging devices record read/write operations on master partitions. If a dirty read/write occurs, the Transaction device alerts Solaris 9 so the read/write can occur again.

SVM supports Device ID's. These let a device keep its configuration even if it is moved from one location to another. With Solaris DiskSuite, if a disk was moved to a different location (`/dev/dsk/c0t2d0` to `/dev/dsk/c0t3d0`) the metadevice databases would become confused.

SVM also supports what are known as "soft partitions." A soft partition is a subdivision of a RAID volume. With a soft partition, more than eight partitions can exist on a single hard drive.

SVM is now integrated into the Solaris Management Console under the Extended Volume Support Icon.

New Install Methods

One of the best improvements in Solaris 9 is its new installation methods. Solaris Live Upgrade is a dramatic improvement to the "destroy and rebuild" process of installing Solaris. If something goes wrong, the system administrator can restore a server back to its original condition, so that performance is not affected by a bad operating system upgrade.

Solaris Live Upgrade

In the past, to install a new version of Solaris, system administrators started by removing the drive with the "good" version of Solaris from the server. Then, they would rebuild the server with the new Solaris upgrade. Obviously, the server had to be down for a long time during this process.

Solaris Live Upgrade is a very unique feature of Sun Solaris. The currently running version of Solaris is duplicated on another disk. The *duplicate* can then be upgraded with software packages and patches. The next time the system is rebooted, the duplicate is run for the first time. The original operating system is not harmed. If there is problem with the new operating system, the server can be rebooted again and the older version of Solaris can be used until the duplicated system is fixed. This eliminates the chances of creating a bad server, as well as eliminating the need for substantial server downtime.

Web Start Flash

Solaris Web Start Flash lets a system administrator install Solaris 9 on a server or workstation. That Solaris workstation or server can then be used as a reference to create what is known as a Web Start Flash Archive. The archive is then used as a replication image to other servers and workstations. The archive can be installed via JumpStart (set up on the new server via FTP from a share point on the network.) or Solaris Live Upgrade.

Other Interesting Improvements

This last section is a catch-all category of different improvements that have been added to Solaris 9. Understand that Sun Microsystems constantly makes improvements to the Solaris 9 operating system, so some of these referenced changes may be updated sometime in the near future.

Netscape Navigator

Under an alliance between Sun Microsystems and Netscape, Solaris 9 comes with the Netscape 6.2.1 browser (as of the publication date) . This browser supports the following protocols:

HTML (Hyper Text Markup Language)	Commonly used on webpages.
XML (Extensible Markup Language)	Allows for adaptable and flexible information identification and document assembly by using user-specified tags and metadata. It is more powerful and configurable than HTML.
LDAP (Lightweight Directory Access Protocol)	An industry wide standard protocol used to access directory services. LDAP can be used to store almost any company-wide form of information, such as phone numbers, email directories, and team schedules. LDAP can be thought of as an organization's database.
DOM (Document Object Model)	A model by which a document or webpage contains various objects (links, elements) that can be dynamically manipulated.

CSSLv1 (Cascading Style Sheet Level version 1) A style sheet mechanism that lets HTML authors add stylistic elements such as fonts, spacing, and colors to HTML documents.

Netscape Navigator 6.2.1 can be used to read email, browse webpages, and receive instant messages. Future versions of Netscape Navigator for Solaris can be obtained directly from Sun Microsystems at <http://www.sun.com/solaris/netscape>.

Microsoft's Internet Explorer is not available for Sun Solaris. There are other alternative browsers, such as Opera and Mozilla, that can be used with Sun Solaris.

iPlanet LDAP Directory Server Integration

The iPlanet LDAP directory is included with Solaris 9. This LDAP server can be used with any LDAP application, such as email systems and HR records. The **idsconfig** utility is used to set up and configure the iPlanet LDAP directory. The **ldapaddent** utility is used to add information to the LDAP database.

After the LDAP server is set up, administrators can manage the server through the Netscape Console. This lets administrators perform all the common LDAP functions, such as creating access rights, replicating of the LDAP database, and so forth. The iPlanet Directory Server supports Attribute Mappings and Service Search Descriptors. The LDAP server is compatible with LDAP clients created with the LDAP Software Developer Kits for C and Java. The LDAP server also supports very strong authentication with TLS encrypted sessions.

Unfortunately, the iPlanet Directory Server is not free. After a short trial period, a company needs to pay to continue using it. The good news is that the license fee is very small (\$2.50 per connection) so small and midsized companies can also use the LDAP server.

SunScreen 3.2

SunScreen 3.2 is a stateful firewall that is included with Solaris 9. This firewall can be set up in stealth mode, so that no visible IP address is associated with the firewall (harder for hackers to penetrate).

The SunScreen 3.2 firewall has both a GUI interface and a command line mode. The command line mode is very useful if a mistake is made when setting up the firewall and Ethernet traffic is disabled. The system administrator can then come in through the terminal port and manage the SunScreen firewall through a terminal session. SunScreen 3.2 supports background NAT (Network Address Translation). It also has built-in support for IPsec, IKE, SKIP and VPN client support.

Solaris Freeware

There are two easy places to get Solaris freeware. The official Sun repository is at <http://www.sun.com/software/solaris/freeware.html>. You can also get GNU software at <http://www.gnu.org>. Please note that this software is not pre-compiled for the Solaris operating system.

Another terrific internet location for Solaris software is <http://www.sunfreeware.com>. This is the best place to get pre-compiled Solaris software. Although it is not an official Sun site, it does have Sun Microsystems sponsorship. This site compiles Solaris freeware so that it can simply be installed into a Solaris server without having to mess around with compilers and libraries.

End of NIS+

Sun Microsystems will discontinue support for NIS+. Networks will have to move to LDAP servers for client-requested information. For the latest information on the transition from NIS+ to LDAP, see <http://www.sun.com/directory/nisplus/transition.html>. Take this announcement with a grain of salt. LDAP is very complex, and in the opinion of the author, there will be enough back pressure against this decision that Sun Microsystems will continue NIS+ in the future. LDAP is just too complex for small companies to work with. NIS+ is a needed component in a company's server management toolbox.

DVD Installation Disk

Solaris 8 was only distributed on CDROMs. Solaris 9 now comes in a convenient DVD disk format that includes all the software found on the Solaris 9 Install CDROM, Solaris 9 Software 1 of 2 CDROM, Solaris 9 Software 2 of 2 CDROM and the Solaris 9 Languages CDROM. This is very helpful when it comes to creating a server in a remote location. If a DVD drive is installed on the server, the Entire Distribution plus OEM can be installed on that server. This eliminates the need to have someone change CDROMs during the installation.

Solaris Web Start Wizards SDK 3.0.1

The Solaris Web Start Wizards are used to create wizards for the installation and setup of both Solaris and Windows versions of a Java program. Non-Java programs are also given limited support with the Solaris Web Start Wizard SDK (Software Development Kit).

Graphical Workspace Manager Upgrade

The Graphical Workspace Manager has been updated. This program provides a graphical representation of workspaces. Workspaces can be changed by the click of a mouse button. This upgrade increases the number of

workspaces that can be displayed in the CDE. An Options dialog box has been added to let the user select different display options. Applications can be added to and removed from workspaces by dragging and dropping them between workspaces.

Software Development Libraries

The following new software libraries are included with Solaris 9:

- Glib 1.2.10** A collection of data types, type conversions, macros and string utilities, and a lexical scanner.
- GTK+ 1.2.10** Taken from the GIMP toolkit. This is a set of libraries that are used to create GUI displays.
- Jpeg 6b** The popular JPEG picture format for images.
- Libpng 1.0.10** The PNG reference library to make PNG images.
- Tcl/tk 8.33** A TCL TK GUI toolkit that is used with the TCL scripting languages.
- Libtiff f3.55** A library for use with the TIF image format, plus tools for working with TIF images.
- Libxml2 2.3.6** Used for XML support.

WBEM

Web Based Enterprise Management 2.5 is a Web-based management tool for enterprise management. SNMP (System Network Management Protocol) is supported by WBEM. WBEM sends out standard SNMP information to SNMP management tools, such as HP Openview. The SNMP adapter maps SNMP requests to CIM (Common Interface Model) WBEM instances and properties. This includes the class name, Abstract Syntax Rotation One (ASN.1) and the property name for each object. The CIM Object Manager listens on TCP port number 5987 for Remote Method Invocation connections. HTTP port connections are listened for on TCP port number 5988.

Improved Solaris Management Console

Solaris Management Console (SMC) is a convenient centralized GUI console that is used to manage a wide range of system administration tasks. Among the functions that system administrators can perform with SMC are:

- Log Viewer** Look at various system logs from the operating system and other applications.
- Performance** Track the use of system resources, such as CPU time and disk space.
- Patches** Examine the patches that are present on other systems, upgrade or remove patches from the SMC.
- Disks** Use the Sun Volume Manager within the Sun Management Center.
- Serial Ports** Configure serial ports for terminals and modems. Much simpler than using command line port SAF/SAC commands.
- Mount Points** Easily create share points and mount points on different systems. Much faster than logging into a server to create the mount/share points by hand.
- Schedule Jobs** Schedule jobs for later run times. Much easier to use than the **cron** and **at** commands.
- Users and Groups** Set up RBAC (Role Based Access Control) for users and groups with a GUI display. Users and groups can also be set up using templates for users that are similar in job duties. Mailing lists can also be set up from the SMC.

System Information	Gather information about other servers on the network. The information includes information about the system's hardware, software packages and the operating system itself.
Diskless Clients	Manage diskless clients with SMC command line tools.

Human Readable Output

The **du**, **ls**, and **df** commands now have a **-h** option to display output in a format that humans can read. The traditional output from these commands was in blocks (½ kilobytes) or kilobytes. When a directory is four GB in size, reading its size in blocks is very difficult. The output can be displayed in kilobytes, megabytes and terabytes if the file or directory is greater than 1 KB in size.

Improved Software Support

In the past, developers could only work with 8 KB memory pages. Now, MPSS (Multiple Page Size Support) lets developers use any size memory page to access virtual memory. This feature is very useful for database applications that are very memory-hungry. The small 8 KB memory pages can be replaced with larger pages that improve memory I/O functions.

Another software support improvement is with the **pkgmk** utility. This creates packages with longer names (32 characters) instead of the very short 8-character package names.

Extra Value Software

There are two types of software included with Solaris :

Co Bundled - Software that was previously shipped separately.

Early Access - Evaluation software that is not supported by Sun support contracts.

Some of these software packages are located in the Solaris 9 Software 2 of 2 CDROM. Other software packages are located in the additional CDROMs that are shipped with the Solaris System Administrator's Media Kit.

Freeware

The license for each package should be in the directory **/usr/share/src/<name_of_the_freeware_package>**.

By definition this software should be free. The only area of concern comes when these freeware utilities and programs are added to commercial software programs.

Apache 1.3.20	Free webserver used to display HTML pages
bash 2.50	Bourne-shell-compatible command interpreter
bzip2 1.0.1	Block sorting file compressor
grep 2.4.2	Pattern matching program
gzip 1.3	GNU version of the zip utility
less	A command that is similar to the more command.
mkisofs	Used to make ISO9660 file systems for CDROM burners.

tar 1.13	File compression utility.
Samba 2.2.2	Free SMB and CIFS server and client for use between UNIX and Microsoft Windows and other operating systems that use SMB and CIFS. This lets unrelated operating systems share printers and file resources.
tosh 6.0.10	A C shell with advanced features like file name completion and command line editing.
zip 2.3	The common zip program to make .zip files.
zsh 3.0.8	A shell used for shell script command processing and as an interactive login shell command interpreter.
Tcp Wrappers 7.6	A small program that filters requests for network services. The program logs the client's name that made the request. This is a good security tool.

As mentioned above, Solaris Freeware can be downloaded from the following website:

<http://www.sun.com/software/solaris/freeware.html>

As mentioned above, pre-compiled software can be downloaded from the excellent website:

<http://www.sunfreeware.com>

Key Points to Remember

This chapter introduces Solaris power users to some of the newest features of Solaris. Most junior level system administrators should spend their time learning the basics of Solaris before trying to tackle advanced Solaris features. A company is not going to let a junior level system administrator work on the LDAP server or DNS server.

Chapter 21 JumpStart Install Methods

Lessons in This Chapter

Lesson 21.1 Working with the <code>re-preinstall</code> Script	21-3
Lesson 21.2 Create the Profile Diskette.....	21-7
Lesson 21.3 Creating a Valid <code>rules</code> File	21-10
Lesson 21.4 Creating a <code>profile_text_file</code> and a <code>rules.ok</code> File	21-16
Lesson 21.5 Using <code>pinstall</code> for a Local JumpStart Installation.....	21-18
Lesson 21.6 Constructing a <code>sysidcfg</code> File	21-19
Lesson 21.7 Performing the JumpStart Installation.....	21-19
Lesson 21.8 Copying Setup Scripts from the <code>tools</code> Directory.....	21-23
Lesson 21.9 Using <code>setup_install_server</code> Script with the <code>-b</code> Option.....	21-25
Lesson 21.10 Using <code>add_to_install_server</code> with the <code>-b</code> Option.....	21-25
Lesson 21.11 Sharing a JumpStart Directory	21-26
Lesson 21.12 Updating The <code>rules</code> File.....	21-26
Lesson 21.13 Perform the JumpStart Installation	21-31

Introduction

Solaris 9 features five different ways to install the operating system. These are:

Web Start

This is a very common method of installing Solaris 9. If a monitor is attached to the system and a frame buffer is in the system, the Web Start installation will be started and displayed in GUI (Graphical User Interface) form. If graphics capability is not available, a text version of the Web Start installation will be presented. Both methods provide the same installation questions.

The Web Start installation requires user intervention. It is basically an installation wizard with graphical windows that present the user with questions. It is the easiest installation method to use because definitions and help screens are shown during the installation. This install method was covered in Chapter 2.

Factory JumpStart

The Factory JumpStart method makes it easy to install the default Solaris 9 operating environment on a brand new workstation or server that does not have an operating system installed. Simply power up the system, insert the Solaris 9 CDROM software 1 of 2 (or the Solaris 9 DVD), and type the command `boot cdrom` at the OK prompt. This only works for new systems that do not have an operating system installed. This installation puts most of the available hard drive space in the `/export/home` directory.

If an operating system already exists, and you want to reset the operating system to the original factory defaults, you can type `boot cdrom -s` and then use the `re-preinstall` command to install the original factory software on the system's default boot disk. This can be slightly customized for different hardware. Lesson 21.1 covers the `re-preinstall` command.

- Custom JumpStart** The Custom JumpStart installation lets a system administrator set up configuration files that automate the installation of Solaris 9. Once the configuration files are created, the system administrator starts a JumpStart installation and then ignores the server until the installation is complete. This is very useful when a company needs to set up a large number of servers. The installation can be done from a local CDROM and floppy diskette or over a network connection (requires an Install server on the same network section). Customized pre-installation and post-installation shell scripts can be added to the JumpStart installation. The JumpStart installation is covered in detail in this chapter.
- Web Start Flash** The Web Start Flash installation is used to create a copy of a known good server. This good server is known as the “master server.” After the master server is checked and tested, it is used as the source for a Web Start Archive. This archive contains all the files that reside on the master server. This archive can be used on a new empty server. All the files from the master server are copied to the empty server. The new server is known as a “clone” of the master server.
- Live Upgrade** The Live Upgrade installation is new in Solaris 9. With this feature, the operating system on any workstation or server with a second hard drive can be upgraded while the original operating system is being used. After the new operating system is installed, the server or workstation is gracefully brought down to the **OK** prompt. The OpenBoot environmental variable **boot-device** is set to the second hard drive. The command **boot** is typed at the OpenBoot **OK** prompt. The new operating system is then run. If problems arise with the new operating system, the previous system disk can be booted to restore server or workstation functions until another Live Upgrade installation is performed. This prevents the usual problems with upgrades, server down time and the possibility of losing a good server.

The re-prinstall Script

The **re-preinstall** script is used for the Factory JumpStart installation method. This section covers the use of this script.

The **re-preinstall** script can be used to set up the original factory JumpStart files on the default boot disk, or on another disk. These files can then be used to install the OS on a new or old system.

The **re-preinstall** script supports the following options:

re-preinstall <options>

- k <platform_name>** The architecture of the hardware. This information can be found using the **uname -i** command. For example, some valid names include **sun4u**, **SUNW** and **Sun-Blade-100**.
- m </Absolute_path/Tools/Boot>** On the Solaris 9 Software 1 of 2 CDROM there is a directory named **/Tools/Boot**. If the CDROM is copied to an alternate location, then the absolute path name to the **Solaris9/Tools/Boot** directory must be provided. If the Solaris 9 Software 1 of 2 CDROM is in the CDROM drive, this option is not necessary.
- <target_slice>** Use this option if the root slice of the target hard drive is not installed at the default **c0t0d0s0** location. A different device path can be specified.

Lesson 21.1 Working with the re-preinstall Script

This lesson will not harm your current operating system. It simply shows the results of using the **re-preinstall** script. In this lesson, a Preinstall Boot Image is dumped to the second hard drive. This image contains the Preboot Configuration files, which are used during a Factory JumpStart installation. Readers can examine these files and become familiar with them.

WARNING

Warning – This lesson will destroy any data in the `c0t2d0s0` slice (the second hard drive). Make sure that all critical data from the second hard drive is saved before performing this exercise.

1. Log in as the root user.
2. Insert the Solaris 9 Software 1 of 2 CDROM.
3. Open a Terminal window.
To open a Terminal window, right click anywhere in unoccupied desktop space, left click on the Tools menu item, left click on the Terminal icon.
4. Type the command `cd /cdrom/sol_9_sparc/s1/usr/sbin/install.d`
5. Type the command `re-preinstall -m /cdrom/sol_9_sparc/s1 c0t2d0s0`
Depending on the path variable, the command might have to be typed
`./re-preinstall -m /cdrom/sol_9_sparc/s1 c0t2d0s0`

The device name `c0t2d0s0` is the first slice on the second disk. This command may need modification, depending on the reader's hardware. Use the `format` command to find the device name of the second hard drive.

The terminal screen should now display a message like:

Beginning Preinstall Boot Image (PBI) installation on c0t2d0s0

- Verifying that there is sufficient disk space on c0t2d0s0

- Creating BPI root file system...

- Mounting PBI root file system...

- Copying BPI root image...

2637920 blocks

re-preinstall: Copying selected /usr files to PBI

re-preinstall: Copying /usr/platform to PBI

56688 blocks

Installing PBI boot block (SUNW,SUN-Blade-100)...

Unmounting PBI root image...

Preinstall Boot Image installation complete

A Preinstall Boot Image has now been copied to the second hard drive.

6. Log in as the root user again (if necessary).
7. Open a Terminal window
8. Type the command `mkdir /2nddrive`
9. Type the command `mount /dev/dsk/c0t2d0s0 /2nddrive`
10. Type the command `cd /2nddrive`
This command shows the contents of the Preinstall Boot Image. Understand that this is not a functional operating system. It is simply an OS image that can be used by the Factory JumpStart installation method, which is covered in the next section.

Using the Factory JumpStart Install Method

This section describes how to use the Factory JumpStart installation method to install or reinstall the original factory default installation of Solaris 9 on a workstation or server. This installation method is preferred for Sun resellers who want to make sure the workstation or server they sell has all the factory defaults loaded on the operating system.

Note: Unfortunately, the default Solaris installation has a bad habit of putting almost all the disk space in slice 7 (the `/export/home`) slice. If you don't want this to happen, consider choosing one of the other installation methods.

There are two circumstances where the Factory JumpStart installation can be used.

For a brand new SPARC system *without an operating system*, follow the instructions below:

1. Power on the system.
2. Insert the Solaris 9 CDROM software 1 of 2 CDROM (or the Solaris 9 DVD).
3. Type the command `boot cdrom` at the **OK** prompt.
4. This installation may ask for the system's hostname, IP address, and other very basic information. Follow the onscreen prompts.

For a SPARC system that *already has an installed operating system*, follow the instructions below:

1. Power on the system.
2. Insert the Solaris 9 CDROM software 1 of 2 CDROM (or the Solaris 9 DVD).
3. Type the command `boot cdrom - s` at the **OK** prompt.
4. Type the command `cd /cdrom/cdrom0/Solaris_9/Tools/Boot/usr/bin/install.d`
5. Type the command `./re-preinstall c0t0d0s0`
6. This installation may ask for the system's hostname, IP address, and other very basic information. Follow the onscreen prompts.

Using the Custom Jumpstart Installation Method

The Custom JumpStart installation method requires a Sun SPARC workstation or SPARC server. The jumpstart software is in the Solaris 9 Software 1 of 2 CDROM. This CDROM contains the `/Solaris_9/Tools` directory. This directory contains various files used for a Custom JumpStart.

Common Custom Jumpstart Terms

Before discussing the Custom JumpStart installation any further, it is important that the reader understand some basic terms:

JumpStart Client	This is a server or workstation that does not typically have an operating system installed. The JumpStart client can install the Solaris 9 O.S. from a local CDROM or diskette, or from a JumpStart server over a network connection.
JumpStart Servers	There are three different type of JumpStart servers. For example, one server can contain configuration files while another server contains copies of the CDROM images. There are three types of JumpStart servers, as outlined below.
Configuration Server	This server has the rules and profile text files. When a target system starts a network-based JumpStart installation, it looks for a Configuration server for instructions on what partitions to create and what software to load.

Boot Server This server is used for a client that does not have an operating system. The command **boot network** is typed at the **OK** prompt in the client machine. The Boot server supplies the client with bootup information, such as an IP address and subnet mask. This server needs to be on the same network segment as the JumpStart client so that it can receive the **bootp** packets from the client.

Install Server The Install server contains a copy of the Solaris 9 CDROM or DVD images. This server receives a request from a JumpStart client for an install. The server should be on a network segment that can handle large file transfers. This server generates a lot of network traffic.

Custom JumpStart Scripts and Files

This section only gives a quick review of the most important scripts and files used with a Custom JumpStart installation. These scripts and files will be covered in greater detail later in the chapter.

Rules file The rules file is a text file that holds rules for each group of systems, based upon various physical characteristics (such as type of server, or Ethernet address) The rules file links a group to a corresponding profile.

Profile file The profile file is a text file that determines how the operating system is to be installed for each group. The groups are defined in the rules file.

Check script This script checks the rules file to make sure that the syntax used in the rules file is correct. If the syntax is correct, the check script creates a file named **rules.ok**. This file is used by the JumpStart installation for reference information.

rules.ok file The **rules.ok** file is created after the check script found no errors in the rules file.

setup_install_server

This script creates the Install server. This script is located in the Tools directory in the Solaris 1 of 2 Software CDROM.

add_to_install_server

This script is found on every Solaris 9 CDROM that can have its software spooled on to the JumpStart installation server's spool directory.

sysidcfg A Network JumpStart installation usually requests that a NIS or NIS+ server be present on the network for the JumpStart installation to occur. The **sysidcfg** file is used when a NIS or NIS+ server is not present on the network..

profile diskette

A floppy disk that contains the files necessary to install Solaris 9 on a standalone workstation or server.

group A collection of servers and workstations that need to be set up in a similar manner. A group can be defined by the type of machine, such as E-250 or E-450.

Understanding Jumpstart Installations

A JumpStart installation can be performed locally or over a network. If a server or workstation does not have a floppy drive, the installation needs to be run over a network. A Local JumpStart installation can use the local CDROM drive as the source of the software. If more than one CDROM is needed for the installation, someone will have to be present to change the CDROM disks. If a DVD drive is present, no media changing needs to occur.

For a network installation, three types of servers must exist: a JumpStart Install server, a JumpStart Configuration server and a Jumpstart Boot server. These types of servers can reside on one system or they can be distributed among several systems. For example, the JumpStart Install server can be hosted by serverA and the JumpStart Boot and JumpStart Configuration server can be hosted by serverB.

The Install server is the first server that must be created. The script `setup_install_server` is used to create the Install server.

When the JumpStart client is powered on, the system administrator first needs to bring the system to the **OK** prompt if the `auto-boot?` variable is set to `true`. This is done by pressing the `STOP + A` keys simultaneously on a Sun Keyboard or `CTL + Break` keys on a PC keyboard. The administrator then inserts the Solaris 9 Install CDROM and then types `boot cdrom` at the **OK** prompt for a Local JumpStart installation.

If the JumpStart installation is local, the installation software will check the floppy drive for a valid `rules.ok` file and matching profile files. Not that a JumpStart server or workstation must have a floppy disk drive for a Local JumpStart installation. If a floppy drive is not present on the server, a Networked JumpStart installation is the only other JumpStart option.

For a Network Jumpstart installation, the workstation is gracefully brought down to the **OK** prompt. The command `boot net - install` is used to start a Network JumpStart installation. A JumpStart Boot server must exist on the same network segment as the JumpStart client. The JumpStart Installation Server and JumpStart Configuration server can be on a different network segment.

The JumpStart client uses the JumpStart Boot server's information to obtain basic network information, such as an IP address, hostname, and subnet mask. The Boot server directs the client to the location of the Configuration server and the Install server. These servers can be separate SPARC machines, or they can all reside on the same machine. Once a viable connection has been established between the JumpStart client and the Boot server, the `SunInstall` program runs to begin the JumpStart installation.

Keep in mind that creating a JumpStart installation is a rather time-consuming process in the beginning. The JumpStart installation is very sensitive and will fail if it encounters even a small error, such as a misspelled word or a variable not used in the `sysidcfg` file. Even if all the configuration files are correct, there are times when the intended target server will not match all the rules criteria. If the target server type changes, the configuration files will need to be re-written and checked again.

Quick Tip

- The only sure way to check a JumpStart installation is to perform the installation and then test the results on a test server.
- Creating JumpStart files that don't have syntax errors and that produce the desired results should take an average system administrator about one or two days.
- The author of this book spent two days to produce all the JumpStart materials and examples in this chapter.

Performing a Local JumpStart

A Local JumpStart is a good option when several workstations or servers that are not connected to a local network need to be set up in a similar configuration. Each machine must have a diskette drive. A disk diskette is used, which holds the configuration files: **rules**, **rules.ok**, **profile-text-file**, and **sysidcfg**.

Local JumpStart Steps

This section gives a general outline of the steps needed to perform a Local JumpStart. Don't worry about the details right now. They are covered in the lessons in this chapter.

1. Modify the **rules** file. Create a rule for each group, server, or workstation. A group can be any distinguishing characteristic of a group of machines that should be set up in a similar manner. A rule can also be created for a single machine that needs a specific configuration.
2. For each rule defined in the rules file, create a matching profile text file. A profile text file specifies what software and slices will be created. This file can be named anything. For example, **myprofile1**, **mydog** and **hello** are all valid profile text file names. The only catch is that the file names must be defined in the **rules** file.
3. Test each profile with the **pfinstall** command.
4. Use the **check** command to make sure the **rules** file is good. If the **rules** file is good, a file named **rules.ok** will be created.
5. Create a text file named **sysidcfg** that answers some questions not covered by the profile text file and **rules** files.
6. Copy the profile text files, and the **rules**, **rules.ok** and **sysidcfg** file to a diskette, which we will call the *profile diskette*.
7. Insert the Solaris 9 Software 1 of 2 CDROM in the CDROM drive.
8. Type the command **boot cdrom - install**

Lesson 21.2 Formatting the Profile Diskette

A profile diskette is needed to hold the profile text files and the **rules** and **rules.ok** files. This disk must be formatted with the UFS file system. The UFS file system is the native Unix File System that Solaris 9 recognizes when it starts a JumpStart installation.

This lesson shows the reader how to create the profile diskette. Use a diskette that is in new condition (preferably new). Make sure that it is a High Density 1.44 MB 3-1/2" diskette. The brand does not matter.

The first four steps show how to create this diskette from the command line. Unfortunately, Solaris 9 has terrible command line floppy support. The rest of the lesson shows how to format a diskette from the File Manager GUI. You can format the diskette either way.

Note: Remember that formatting a diskette erases all information on it. If you use a diskette that contains data, back up that data before following this lesson.

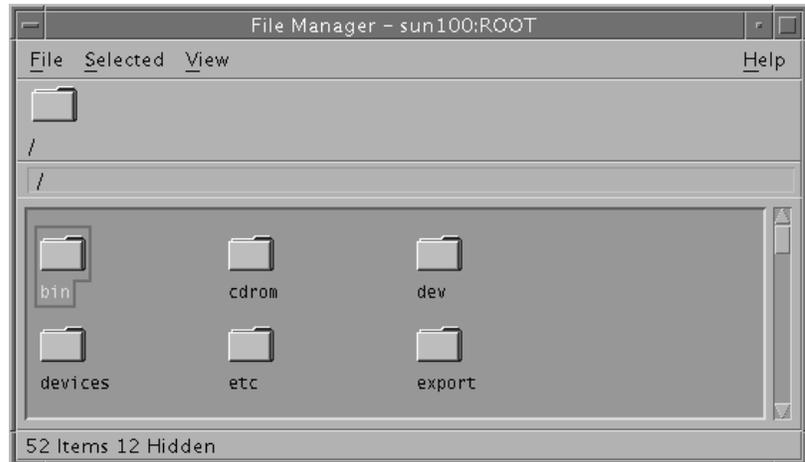
1. Log in as the root user.
2. Open a Terminal window.

3. Insert the diskette to be formatted.
Most users will prefer to format this diskette using the File Manager. If you want to format it from the command line, follow the next step. Otherwise, skip to step 5.
4. To format a diskette from the command line:
 - a. Type the command **fdformat -U**
*The **fdformat** command formats the diskette with the 1.44 MB high capacity file system.*
 - b. Type the command **newfs /vol/dev/aliases/floppy0**
*The **newfs** command creates the file system. If for some reason the floppy is not created successfully, the **newfs** command will generate error messages.*
 - c. Type the command **eject floppy**

If a floppy was created successfully with this example, you can skip the rest of this lesson.

The following steps show how to format the profile diskette from the File Manager.

5. Open the File Manager application.
To open the File Manager application, right click anywhere in unoccupied desktop space. In the Workspace menu, left click on the Files menu choice, then left click on the File Manager icon.
6. In the File Manager, left click on the File menu bar item.
7. Left click on the "Open Floppy" menu choice.
8. When a new window opens, left click on the File menu bar item.
9. Left click on the "Removable Media Manager menu choice.
10. When a new window opens, right click on the floppy icon.
11. In the popup menu, choose Format.
12. In the File System Type window, select the UFS (UNIX)file system, leave the Volume Label blank.
13. Left click on the Format button on the lower left corner.
Formatting a floppy diskette is a slow process. It could take several minutes.
14. When the formatting is done, click **OK** in the Format Complete window.
You will see a display of the floppy's contents, usually the **lost+found** directory.
15. Exit from the File Manager, if desired.



Understanding the rules file

One of the first thing the install program looks for is the file named **rules.ok**. This file is created by the **check** command. The **check** command reads a text file named **rules**. The **rules** file creates a separate rule for each type of SPARC system.

For example, the first rule can be for E-250 servers, and the next one could be for E-450 servers. These rules can distinguish among servers based on characteristics such as the system's disk size, the Kernel, the current operating system and so forth. Each of these rules points to a profile file.

The **rules** file can contain three types of lines. They are:

- Commented lines Must start with the pound (#) symbol.
- Blank lines Have nothing on the line.
- Lines that contain rules. Follow the format described below.

The rules must be based on the following line format:

<keyword> <value> <begin_script_name> <profile_file_name> <end_script_name>

<keyword> Understand that these keywords apply to all kinds of situations, such as new machines, networked machines, and workstations with a current operating system. Some of these keywords obviously won't apply to a local installation of JumpStart.

The keywords are:

any	Anything matches this value
arch	Any system with the specified architecture
domainname	Current operating system's domain
disksize	Disk device name or disk size in megabytes.
hostname	Host name
installed	Current version of Solaris
karch	Kernel architecture
memsize	Total megabytes of physical memory
model	Model number
network	IP address.
totaldisk	Total disk size in megabytes

< value > The <value> option supports several logical operators, such as:

.....

<!>	not
<&&>	and

<begin_script_name> The system administrator can run any available command with a Bourne shell script before the installation begins.

<profile_file_name> The profile text file. This file can have any name desired, such as: **sun250**, **myprofile**, and **mydog**. The profile text file is used when a <keyword> match is found. The profile file is described later in the chapter.

<end_script_name> The system administrator can run any available command with a Bourne shell script after the installation is done.

A sample rules files can be found in the following directory on the Solaris 9 Software 1 of 2 CDROM:

/cdrom/cdrom0/s0/Solaris_9/Misc/jumpstart_sample

Understand that the **SunInstall** program will go down each line in the **rules** file. When a match is found, it will read the matching **profile text file** associated with the rule. At the bottom of a typical rules file is the keyword **any**. This keyword is a catch-all category that will match any computer or system. Always put the **any** keyword at the bottom of the file, or none of the other keywords will work. Remember that as soon as a match is found, the rest of the **rules** file is ignored. What doesn't match with **any**?

Lesson 21.3 Creating a Valid rules File

In this lesson, readers will create a **rules** file from the sample rules files that are copied from the **jumpstart_sample** directory. This **rules** file will work with any architecture or system. Readers should practice creating **rules** file for the particular servers or workstations that they will be using.

1. Log in as the root user.
2. Open a Terminal window.
3. Insert the Solaris 9 Software 1 of 2 CDROM
4. Type the command **mkdir /JumpStart** (if the directory does not exist).
5. Type the command **chmod 777 /JumpStart** (if the directory was just created).
*The **chmod 777** command changes the directory permission to read/write/execute for everyone.*
6. Type the command
cp -r /cdrom/cdrom0/s0/Solaris_9/Misc/jumpstart_sample/* /JumpStart
*If this command is too difficult, just **cd** to the **jumpstart_sample** directory, then type the command **cp * /JumpStart***
7. Type the command **cd /JumpStart**
8. Type the command **cp rules rules.backup**
9. Type the command **vi rules**
10. Edit the **rules** file. Use the pound symbol (#) to turn every line into a comment (**This is important → Do this for every line**).
*Read the **rules** file. This is very well documented file.*
11. Add this line at the end of the file

```
any - - myprofile
```

(the number of spaces is not significant).

*Note: If the **rules** file becomes hopeless damaged, and you need to return to the original **rules** file, you can restore it in either of the following ways:*

- If you copied it to **rules.backup** in step 8, just type the command **cp rules.backup rules**
- You can copy it back from the CDROM. The filename is

```
/cdrom/cdrom0/s0/Solaris_9/Misc/jumpstart_sample
```

Understanding a Profile Text File

A profile text file does not have the name “profile text file.” It can be anything, like **myprofile**, **profile32**, **salesprofile**, or even **MyDogLovesBones**.

Let's say that a line in the **rules** files reads:

```
any - - myprofile -
```

The SunInstall program will look for a profile text file named **myprofile** in the same directory as the **rules** file. If a line was added that read:

```
any - - MyDogLovesBones
```

The SunInstall program will look for a profile text file named **MyDogLovesBones** in the same directory as the **rules** file.

The profile text file directs the installation of the system. Basically, this file can be thought of as an “answer file” for all the installation questions. The installation questions provide the system with information such as the size of a disk slice, what software cluster to install and so forth. When a workstation or server matches the search criteria in the **rules** file, the profile text file is used to answer the installation questions.

In the first example of a **rules** file line immediately above, the rule any matched any workstation or server. So, a file name **myprofile** is called upon to install the software.

The first line of a profile text file must use the keyword **install_type**. Figure 21.1 shows an example profile text file

```
install_type <value> ← All of these values will be described later in the chapter
initial_install
boot_device c0t2d0s0 update
cluster SUNWCreq
dontuse c0t0d0
```

Figure 21.1 Example Profile Text File

Profile File Keywords

This section discusses some of the more common keywords that appear in a profile text file.

backup_media

The **backup_media** is used when a system is being upgraded. If there is not enough space on the backup slice or slices, the JumpStart installation will copy the file system’s contents to a backup device, like a tape or disk drive. The following options are supported:

backup_media <location>

<location>

local_diskette Back up the file system to the local floppy disk. This is not a very good option. Given the size of most hard drives, it’s a rather ridiculous option!

local_filesystem This must be a local file system; it can not be one of the slices that is being upgraded.

/mount_directory A mounted file system in **/etc/vfstab**. This must be a local file system.

/dev/dsk/c#t#d## The contents of the target file system are dumped to a storage file system **/dev/dsk/c#t#d##**.

remote_filesystem This must be a host that gives read/write permission. The hostname can be specified by a hostname or an IP address.

host:/file_system Follows the format of:
<hostname or IP address:/share_directory_on_NFS_server>

remote_system This works just like the **remote_filesystem**, except that the foreign host must allow **rsh** connections. The foreign host must have an entry in the **.rhosts** file.

boot_device

This keyword specifies the location of the boot device. If this keyword is not specified, the JumpStart installation will choose any disk. Usually, the **c0t0d0** disk is chosen.

The following options are supported:

boot_device <location> <update?>

<location>

c#t#d# Specifies a disk with the physical device name.

existing The current boot device will be used, and will be overwritten.

any Uses the current boot device if possible, chooses another one if the existing boot device is not available. This is a dangerous option to use, because the boot device should be defined and not chosen by anything.

<update?>

update Update the OpenBoot variable boot device.

preserve Don't update the OpenBoot variable boot device.

core <corename> The software clusters can be chosen with the keyword clusters. There are five default clusters that can be chosen:

SUNWCXall	Entire Solaris Software Group PlusOEM Support
SUNWCall	Entire Solaris Software Group
SUNWCprog	Developer Solaris Software Group
SUNWCuser	End User Solaris Software Group
SUNWCreq	Core

Cluster

This option adds or removes specific software clusters to the target server. A cluster is a collection of software packages. CDE Developer Software, SUNWcdtdev 1.1 is an example of a software cluster.

Cluster <cluster_name> <add-delete>

<cluster_name> The name of any software cluster in the media. Example:

SUNWbtoox	The CCS libraries
SUNWdpl	The Developer Profiled libraries.

<add-delete>

add Add the software cluster to the installation.

delete Remove the software cluster from the installation.

dontuse

This option is used to tell the JumpStart installation not to use a particular disk. In the following lesson, the **c0t0d0** disk is specified as a “don’t use” disk, so that the original operating system is preserved.

filesys

This option is used to set up the slices on the server or workstation. The default installation of file systems on Solaris 9 is not very good. Almost all system administrators specify a custom file system.

The options for this are:

```
filesys <device_name> <size> <mount_directory> <additional_options>
<mount_options>
```

<device_name>

- c#t#d#s#** The name of the device, such as **c0t2d0s0**.
- any** Put the file system on any disk. **DO NOT USE THIS OPTION!!!**
- rootdisk.s#** Put the file system on the root disk, slice #

<size>

- #** Size of the slice: 200 = 200 MB, 3000 = 3 GB
- all** Use all the disk space for this slice; no other file systems can be created on the disk.
- auto** The size of the file system will depend on the software selected. This is another **DO NOT USE!** option. Specify the size of the slice by hand.
- existing** Use the current size of the slice on the disk. For example, if the current system has a root (/)file system of 2 GB, the new operating system will have a root (/)file system of 2 GB.
- free** Any remaining disk space is used with this slice. This must be the last entry. Any **filesys** line that follows this one will not work.

<mount_directory>

- Mount Point** A defined mount point like / or /opt or /usr or /myslice
- swap** A reserved keyword to make the swap slice.
- unnamed** A raw slice that is not mounted or named.
- ignore** Do nothing to this slice.

<additional_options>

preserve Do not damage the file system specified; can not change the size of the file system. This option can only be used if the size of the slice is set to **existing**.

<mount_options> Any of the special **mount -o** options. For the UFS file system, to view these options, type the command **man mount_ufs**.

geo This option is used to indicate the geographic locale to install with Solaris 9. Some of the more popular options are:

Asia	- Asia
N_America	- North America
S_America	- South America
C_Europe	- Central Europe
E_Europe	- Eastern Europe
N_Europe	- Northern Europe
S_Europe	- Southern Europe
W_Europe	- Western Europe
N_Africa	- Northern Africa
C_America	- Central America
M_East	- Middle East
Ausi	- Australia, New Zealand

install_type

This keyword must be on the first line of the profile text file. It has three possible values::

initial_install This install type cleans off the original operating system and installs a new operating system. It is possible to install an operating system on a different disk than the system disk. Another option is to simply pull the system disk with a known good operating system and install a blank disk. If the JumpStart installation goes bad, simply replace the blank hard drive with the original hard drive.

upgrade Upgrade the previous versions of Solaris. This will only work with Solaris 2.6, Solaris 7 and Solaris 8. This will not work on an Intel system.

flash-install Works with Web Start Flash installations. This will be covered in the Web Start Flash section of this chapter.

locale

This option lets the administrator choose the locale for the system. This option supports many locales for different countries: There are three English-language locales commonly used in the United States.

en_US.ISO8859-1	ISO8859-1 English
en_US.UTF-8	UTF-8 English (U.S.A., Unicode 3.1)
en_US.ISO8859-15	ISO8859-15 English (U.S.A., ISO8859-15 - Euro)

package

This keyword is used to add or remove software packages to the server. Some packages can not be removed. JumpStart will also try some sanity checking when it comes to package dependencies. For example, if a CDE developer package is installed, the CDE must be installed. Do not count on the JumpStart installation to be 100% correct. The best practice is to not remove packages from the common software groups. Adding software packages usually will not damage a server, but removing software packages can produce unpredictable behavior.

This keyword supports the following options:

package <package_name> <additional_options>

<package_name> - SUNWaudio

<additional-options>

add Add the software package to the installation.

delete Remove the software package from the installation.

partitioning

This keyword partitions the disks in the default manner. Any slice explicitly defined by the keyword **filesystems** is set up on the server according to the keyword **filesystems**. The **filesystems** variable **rootdisk** is also recognized by the partition keyword.

The two options are

default The slices will be created, except for the slices specified by the keyword **filesystems**

explicit Preserve all file system layout, except for:

root (/)
openwin (/usr/openwin)
opt (/opt)
var (/var)
user (/usr)

root_device

Specifies the system's root device. The only option is **c#t#d##**

system_type

Specifies the type of system installed. Only two values can be used: **standalone** and **server**. If this variable is not specified, **standalone** is used by default. A standalone server is a server that is not connected to a network (a very rare situation). A server is a system that is connected to a network.

usedisk

This is the exact opposite of the variable **dontuse**. This tells the JumpStart installation what disk should be used for the installation. The variable **usedisk** supports disk labels **c#t#d#**. The keywords **dontuse** and **usedisk** can not be in the same profile.

Lesson 21.4 Creating a Profile Text File and a `rules.ok` File

In this lesson, readers are going to look at the example profiles that are available in the `/JumpStart` directory. The `grep` command can be used to search through different types of profile files. Then, readers are going to make a custom-made profile. This profile will be a very generic profile that will most likely not fail on any system.

WARNING

In this lesson, make sure that any `c0t2d0` references in `filesys` lines in `myprofile` point to the second hard drive (which does not hold the operating system). The device number `s0` points to the first slice. This drive should not contain operating system files.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command `cd /JumpStart`
4. Type the command `cat any_machine`
This command is used to view some of the settings in the profile named `any_machine`
5. Type the command `grep cluster *`
This command searches for all the files with the text "cluster" in them. If there is a profile with the option `cluster`, it will be shown by the `grep` command. Type the command `cat <profile_name>` to view the contents of the profile.
6. Type the command `cp any_machine myprofile`
7. Edit the `myprofile` text file so it looks like this.
Remember that any `c0t2d0` references must refer to the second hard drive, not the system drive.

```
install_type initial_install
system_type standalone
partitioning default
cluster          SUNWCuser
filesys          c0t2d0s0 3000 /
```

8. Add this line to the end of the `myprofile` file

```
filesys          c0t2d0s1 512 swap
```

This line sets the size of the swap partition. As before, make sure the `c0t2d0` device numbers point to the second drive. The number `s1` should be the first slice on a second disk. This disk should not contain an operating system files.

9. Add this line to the end of the `myprofile` file

```
filesys          c0t2d0s7 free /extra
```

This line uses any extra disk space in slice 7 (`s7`) with the keyword `free`. The slice is mounted under a directory named `/extra`. As before, make sure the `c0t2d0` device numbers point to the second drive. The number `s7` should be the seventh slice on the second drive. This drive should not contain any operating system files.

10. Add this line to the end of the `myprofile` file

```
fileys      c0t2d0s7 free /extra
dontuse     c0t0d0
```

*This line makes sure not to use the **c0t0d0** system disk.*

11. Save the file as **myprofile**

*The final **myprofile** file should look like this:*

```
install_type initial_install
system_type standalone
partitioning default
cluster      SUNWCuser
fileys       c0t2d0s0 3000 /
fileys       c0t2d0s1 512 swap
fileys       c0t2d0s7 free /extra
dontuse      c0t0d0
```

12. Type the command **check**

*The **check** command creates a file named **rules.ok**. The screen should display the following output:*

```
Validating rules...
Validating profile myprofile...
The custom JumpStart configuration is ok.
```

13. Type the command **ls**

*The file **rules.ok** should be in the **/JumpStart** directory.*

14. Insert the formatted floppy disk in the floppy disk drive.

*If the floppy diskette is not seen, type the command **volcheck**.*

15. Type the command **cp /JumpStart/rules /floppy/floppy0**

16. Type the command **cp /jumpstart/rules.ok /floppy/floppy0**

17. Type the command **cp /jumpstart/myprofile /floppy/floppy0**

*The three previous commands copied the **rule**, **rules.ok** and **myprofile** files to the floppy diskette.*

Using the **pfinstall** Utility to Check a Profile

The **pfinstall** utility is used to check a profile. This command basically produces output from a “dry run” installation.

One of the things that the **pfinstall** command needs is a text file that has the output of the **prtvtoc** command. So, before running the **pfinstall** command, run the following command:

```
prtvtoc /dev/rdisk/<c#t#d#s2> > mydiskconfigfile
```

*The file **mydiskconfigfile** will show the partition table of the disk specified on the command line. Here, **/dev/rdisk/<c#t#d#s2>** is the second slice of the hard drive that will receive the JumpStart installation.*

For example, on a SunBlade 100, the command

```
prtvtoc /dev/rdisk/c0t0d0s2 > mydiskconfigfile
```

*should be run before the **pfinstall** command is run (of course, you can use any name for this file).*

The **pfinstall** command supports the following options:

pfinstall <options>

<options>

- c** Path to the software image. This can be the CDROM or a spool directory for the software packages.
- d** Use a disk configuration file. Use the **mydiskconfigfile** (or other filename) that was created above.
- D** Use the current system's disk geometry for the test run.

The command should look something like this:

```
pfinstall -d mydiskconfigfile -c /JumpStart myprofile
```

Lesson 21.5 Using **pfinstall** for a Local JumpStart Installation

In this lesson the **pfinstall** program is used to check the **rules.ok** file and the profile, with a dry run installation.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command **cd /JumpStart**
4. Type the command **cp myprofile myprofile.backup**
5. Type the command **prtvtoc /dev/rdisk/c0t0d0s2 > /JumpStart/mydiskconfigfile**
*The **prtvtoc** should be used on slice 2 of the JumpStart target drive.*
6. Type the command **vi runpfinstall**
*You will need to enter the line shown below each time that a new profile file is create. This is a very long line, and you will need to enter it often. This can be very fatiguing. Instead, It's much easier to create a Bourne shell script that will execute this line each time you make a change to the **myprofile** file. Create a Bourne shell script with the following lines. Type the second line (starts with **/usr/sbin/install.d/pfinstall**) as one long line. Do not press the Return key until you have typed the entire line!*

```
#!/bin/sh  
/usr/sbin/install.d/pfinstall -d /JumpStart/mydiskconfigfile -c  
/cdrom/cdrom0/s0 /JumpStart/myprofile
```
7. Save the script as **runpfinstall**
8. Type the command **chmod 777 runpfinstall**
9. Type the command **./runpfinstall**
*Now, play around with the **myprofile** file. Try different commands, then run the **runpfinstall** command several times. After playing with the **myprofile** text file, return the file to its original condition (as it was first created in the previous lesson) by typing the command **cp myprofile.backup myprofile**.*

Understanding the **sysidcfg** File

The **sysidcfg** file answers some of the first questions during the installation that are not answered by the profile text file. For example, the **sysidcfg** command needs to know information such as the time zone, the IP address of the host and the hostname.

If the `sysidcfg` file is not present, the `SunInstall` program will ask the user a series of questions that provide this type of information. If the `sysidcfg` file is present, the `SunInstall` program will only ask the questions that are not answered in the file.

Lesson 21.6 Constructing a `sysidcfg` File

In this lesson, the reader will create a `sysidcfg` file that will answer the questions that the `SunInstall` program needs for a hands-free installation. This file is a text file that has several keyword variables, one keyword per line. In this example, the `sysidcfg` file is created with the `vi` editor and then copied to the system's floppy disk drive.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command `cd /JumpStart`
4. Type the command `vi sysidcfg`
5. Edit the `sysidcfg` file so it looks like this:

```
system_locale=en_US
timezone=US/Mountain
timeserver=localhost
name_service=none
network_interface=primary { hostname=SUN100
                             ip_address=192.168.0.4
                             default_route=192.168.0.1
                             netmask=255.255.255.0
                             protocol_ipv6=no }
security_policy=none
```

6. Save the file as `sysidcfg`
7. Type the command `cp sysidcfg /floppy/floppy0`
This copies the `sysidcfg` file to a floppy disk.
If the floppy is not recognized, type the command `volcheck`, then type the command `cp sysidcfg /floppy/floppy0`

Lesson 21.7 Performing the JumpStart Installation

WARNING

This lesson installs a new copy of the Solaris 9 operating system. Always use extreme caution when performing this type of installation. If you have only one hard drive, and something goes wrong, your workstation will not boot properly.

If you have only one hard drive, use EXTREME CAUTION if you follow this lesson!

If you have two hard drives, you should remove the hard drive that has a known good copy of Solaris 9 before you follow this lesson. That way, if something goes wrong, you can still boot your workstation from the existing boot drive.

This is the final step for a Local JumpStart installation. This lesson finishes the JumpStart installation of Solaris 9 on the second hard drive. For systems with two disks, this installation should not harm the first disk that already has an operating system (but be careful, anyway).

1. Insert the Solaris 9 Software 1 of 2 CDROM in the CDROM drive.
2. Insert the floppy diskette that you have prepared in the floppy drive.
3. Bring the system down to the **OK** prompt in a graceful manner. Use the **init 0** command or the **shutdown -y -g0 -i6** command if the operating system is running.
4. At the **OK** prompt, type the command **boot cdrom - install**
5. The ordinary bootup messages appear. You should then see:

Configuring /dev and /devices
Searching for configuration file(s)...
Using sysid configuration file from local floppy
Search complete.

*If a blue screen appears, that is the SunInstall program asking for information it could not understand from the **sysidcfg** file. If this happens, make sure to answer the questions. Do not just quit the installation. Answer the questions this time to see what questions are tripping up the SunInstall program. Then, edit the **sysidcfg** file until the **SunInstall** program does not ask any more interactive questions.*

It is very rare to have all the JumpStart files correct the first time. Most administrators find their mistakes when interactive questions are asked. Eventually the right words and variables are found, so that the automatic installation is truly hands-free.

Occasionally a syntax error message will appear describing an error, such as:

Syntax error line 8 position 19, default_route 192.168.0.1

If everything works correctly, a blue screen will appear, with the magic text

System identification complete..

Next comes the line:

Generating software table of contents [this may take a few seconds...]

These messages means that the JumpStart installation has worked!

After the Local JumpStart installation is complete, the system reboots and the root password is requested.

6. Make sure that the Caps Lock key is not down.
If the Caps Lock key is down, in the next step you will create an all uppercase password that can cause confusion.
7. Enter the root password, then enter it a second time to confirm that it was typed in correctly.

For systems that are using a CDROM, a window might appear with the message:

Solaris 9 Software 2 of 2 CDROM is required

This depends on the software cluster selected. The window might also have a smaller text message that reads:

Specific media Please specify the media from which you will install Solaris 9 Software 2 of 2 (SPARC Platform Edition)

You will see the following radio buttons.

- CD/DVD
- Networked File System

Click on **CD/DVD**

8. Insert the Solaris 9 Software 2 of 2 CDROM, then left click on the Next > button.

If for some reason the CDROM drive will not open, follow these steps:

- a. Open a Terminal window.
- b. In the terminal window, type the command **eject cdrom**
- c. Insert the Solaris 9 Software 2 of 2 CDROM.

Wait about 10 seconds for the CDROM to spin up. For some reason, the system may not recognize there is a CDROM disk in the drive. If this happens, reinsert the CDROM disk and then press the OK or NEXT> button again. This might have to be done several times for the CDROM to work.

9. Left click on Next >

*At this point in time, additional software packages are being installed. The installation program has modified the **boot-device** OpenBoot variable to automatically boot the new version of Solaris 9.*

Note: The next time the system returns to the **OK** prompt, reset the boot device OpenBoot variable by typing the command

setenv boot-device disk.

*You can also reset the boot device OpenBoot variable with the **eeeprom** command.*

10. After the software is installed from the Solaris 9 Software 2 of 2 CDROM, a window appears with the title "Installation Summary." In this window will be one or more small gray buttons with the label "Detail.."

Left click on one or more of these buttons.

You will see information about the installation that has just been performed.

11. Files that contain detailed descriptions of the installation can be found in the directory **/var/sadm/install/logs/**

To see this information:

- a. Open a Terminal window.
- b. Type **cd /var/sadm/install/logs/**
- c. Type **ls**

You'll see a listing of files such as

Solaris_9_packages_part_2_install.B09220450
Solaris_On_Sun_Hardware_Documentation_install.B09220640

SunATM_5.1_install.B09220645

SunFDDI_PCI_3.0_install.B09220645

- d. Use the **cat** command to examine any of these files.

12. Left click on the Reboot Now button to reboot the system.

In this lesson, the "End User" software group was selected and installed. This has an extremely limited collection of software packages. Don't be surprised if many desktop tools and the man pages for them are not present. The original operating system is still on the system disk and should be unchanged.

Note: If the original system disk was removed to perform the JumpStart installation, follow these steps:

- a. Replace the original hard drive that was used before the JumpStart installation.
- b. Check the value of the **boot-device** variable.
- c. If this value is not **disk**, at the **OK** prompt, type the command

setenv boot-device disk

This will allow the original system disk to boot.

Performing a Network JumpStart

The JumpStart installation can use the local CDROM and a local floppy diskette for the configuration files and installation software base, or it can use one or more network servers. JumpStart servers are used to host the configuration files and the installation software. All of these files can be on one server, or on three different servers.

The servers are:

Install Server Holds the installation software, the contents of the Solaris 9 Software 1 of 2 CD-ROM, the contents of the Solaris 9 Software 2 of 2 CDROM and the contents of the Solaris 9 Languages CDROM. The Solaris 9 DVD can also be used as the source of the installation software.

The Install server must contain all the software for a desired cluster. Table 21.1 shows the CDROMs that are necessary for each type of cluster.

Clusters	CDROM Required
Core	Solaris 9 CDROM 1 of 2
End User	Solaris 9 CDROM 1 of 2
Developer	Solaris 9 CDROM 1 of 2 Solaris 9 CDROM 2 of 2
Entire Distribution	Solaris 9 CDROM 1 of 2 Solaris 9 CDROM 2 of 2

Table 21.1 CDROMs Required for Installations

Configuration Server This server holds the JumpStart configuration files. These include the profile text files and the following named files: **rules**, **rules.ok**, **sysidcfg**, **beginning_script_file**, and **ending_script_file**.

Boot Server This server holds a miniature copy of the Solaris 9 operating system that the client uses during the installation of Solaris 9. The miniature copy of Solaris 9 is not the installed final copy. It is just a utility version that is used during the installation. This server needs to be on the same network segment as the JumpStart clients. A router or network device can not be between a JumpStart client and the JumpStart server.

There are several scripts that can be used to change an existing server into a JumpStart server. These scripts are located in the **/s0/Solaris_9/Tools** directory on the Solaris 9 Software 1 of 2 CDROM disk. Most of these commands are Bourne shell scripts.

The setup_install_server script

The **setup_install_server** script is located on the Solaris 9 Software 1 of 2 CDROM in the **/cdrom/cdrom0/s0/Solaris_9/Tools** directory.

This script can perform one of two functions. The first function is to copy the CDROM contents to a directory on a hard drive. The second function is to copy all the files necessary for a workstation or server to pull the miniature version of Solaris 9 down off the network to boot.

The `add_install_server` script

The `add_install_server` script is used to add the contents of a CDROM onto a spooled software directory. Each CDROM has its own copy of this script. This script copies the contents of the host CDROM to a directory on the hard drive. It is a good idea to spool the CDROM images onto a hard drive because CDROMs are very slow devices by nature.

The `add_to_install_server` script

The `add_to_install_server` script is found on every Solaris 9 CDROM that can have its software packages added to the spool directory of the Install server. It is located in the `/cdrom/cdrom0/Solaris_9/Tools` directory.

This script copies the contents of the host CDROM to a directory on the hard drive. Make sure to only use the `add_to_install_server` script that came with each CDROM disk. Do not copy the `add_to_install_server` script to the hard drive and use it for all the CDROM disks. It is a good idea to spool the CDROM images onto a hard drive because CDROMs are very slow read devices by nature.

Lesson 21.8 Copying Setup Scripts from the Tools Directory

There is a directory on the Solaris 9 Software 1 of 2 CDROM that can be used to help automate the installation of a JumpStart server. This directory contains various sample profiles, a start script, a finish script and the `check` program. The `check` command is needed to produce the `rules.ok` file. It is easier to work with these scripts if they are copied to a directory on the hard drive.

1. Insert the CDROM labeled Solaris 9 Software 1 of 2.
2. Open a Terminal window.
3. Type the following commands:

```
cp /cdrom/cdrom0/s0/Solaris_9/Tools/add_install_client /JumpStart
cp /cdrom/cdrom0/s0/Solaris_9/Tools/rm_install_client /JumpStart
cp /cdrom/cdrom0/s0/Solaris_9/Tools/setup_install_server /JumpStart
```

Different CDROMs have different directories and scripts. Some of the previous `cp` commands might produce an error message. Use the command `find /cdrom -name add_install_client` to find the JumpStart directory on the CDROM. Try to find all three scripts if they exist.
4. Type the command `cd /JumpStart`
5. Type the command `cat add_install_client`
Look at the files in this directory. These files help to automate the creation of a JumpStart server.
6. If the `/JumpStart` directory contains valid configuration files (`rules`, `myprofile`, `rules.ok`), skip this step.

Otherwise, if the `/Jumpstart` directory *does not* contain valid configuration files, type the command

```
cp -r /cdrom/cdrom0/s0/Solaris_9/Misc/jumpstart_sample/* /JumpStart
```

7. Type the command `cd /JumpStart`
8. Type the command `ls`
This shows some sample files used to create rules files, profile files and the `check` command.
9. Type the command `cat any_machine`
This shows some of the settings for the profile named `any_machine`.
10. Type the command `grep cluster *`
This command searches for all the files with the internal text “cluster.”

If there is a profile with the option **cluster**, it will be shown by the **grep** command. Type the command **cat <profile_name>** to view the contents of the profile.

Play around with the **grep** command to try to find all the files that have samples of different keywords.

Chapter21.1 Using the **setup_install_server** Script

This lesson copies the contents of the Solaris 1 of 2 Software CDROM onto the JumpStart server's hard drive. The directory created in this lesson must exist before you run the **setup_install_server** Bourne shell script.

1. Log in as the root user.
2. Insert the Solaris 9 Software 1 of 2 CDROM.
3. Open a Terminal window.
4. Type the command **mkdir /exportsoftware**
5. Type the command **chmod 777 /exportsoftware**

Because this is a test workstation, security and permissions are not a major consideration. For a production server, don't change the directory's permission to 777; use 755 instead.

*The **chmod 777** command gives all users full read/write/execute permissions on the directory **/exportsoftware**.*

6. Type the command **/cdrom/cdrom0/s0/Solaris_9/Tools/setup_install_server /exportsoftware**

The screen should show

Verifying target directory...

Calculating space required for the installation boot image

Copying Solaris_9 Tools hierarchy...

Copying Install Boot Image hierarchy...

This command copies the CDROM. This could take over an hour, depending on the speed of the CDROM drive.

*If an error occurs during the copying, the script can only be run again on an empty directory. If an error occurs, type the command **rm -rf /exportsoftware** to completely remove the directory, then start this lesson again.*

7. Type the command **ls -l /exportsoftware**
8. Type the command **ls -l /cdrom/cdrom0/s0/Solaris_9/Tools/boot**
*Notice how the **/cdromimage** directory on your hard drive and the boot directory of the CDROM (**s0**) look the same? This happens because you are using the **setup_install_server** script to copy the contents of the CDROM onto the hard drive. If you had just copied the CDROM to a directory on the hard drive, the image on the hard drive would not be correct. The **setup_install_server** script takes into account system settings that a copy operation would not be aware of.*
9. Type the command **cd /**
10. Type the command **eject cdrom**

Just as a side comment, look at the following commands and their output:

```
# pwd
```

```
/exportsoftware/Solaris_9
```

```
# ls -l
```

```
drwxr-xr-x 2 root  bin    512 Apr 15 12:23 Docs
drwxr-xr-x 4 root  bin    512 Apr 15 12:23 Misc
drwxr-xr-x 2 root  bin    512 Apr 15 12:23 Patches
drwxr-xr-x 551 root  bin   11264 Apr 15 12:33 Product
drwxr-xr-x 3 root  bin    512 Sep 29 17:02 Tools
```

```
# pwd
/cdrom/sol_9_sparc/s0/Solaris_9
# ls -l

drwxr-xr-x 2 root  bin    2048 Apr 15 12:23 Docs
drwxr-xr-x 4 root  bin    2048 Apr 15 12:23 Misc
drwxr-xr-x 2 root  bin    2048 Apr 15 12:23 Patches
drwxr-xr-x 551 root  bin    71680 Apr 15 12:33 Product
drwxr-xr-x 2 root  bin    2048 Apr 15 12:23 Tools
```

*Notice that the directories in the **/exportsoftware** directory look like the directories from the CDROM?*

The `setup_install_server` script with the `-b` Option

If the `setup_install_server` is used with the `-b` option, the material copied from the CDROM to the hard drive target directory can be used by JumpStart client computers during a JumpStart Network installation. The `-b` option sets up the target directory to contain the files that the JumpStart client needs to boot.

Understand that some JumpStart clients don't have a boot partition or an operating system on the hard drive. These JumpStart clients must to pull the Kernel and device drivers off of the JumpStart Boot server's hard drive.

Lesson 21.9 Using `setup_install_server` Script with the `-b` Option

After a JumpStart client contacts a JumpStart server over the network, it needs to obtain all the bootup files from the JumpStart server. In this lesson readers will create the boot directory. This directory contains the initial files necessary for the client to boot. Understand that JumpStart clients do not use a local copy of Solaris 9. They always pull the miniature version of Solaris 9 off of the Install server and through the network.

The only reason that the `add_install_server -b` command is run again on a different directory in this lesson is so that the reader can look through both the `/jumpstartboot` and `/exportsoftware` directories. Most system administrators would not produce two separate directories.

1. Log in as the root user.
2. Open a Terminal window
3. Insert the Solaris 9 Software 1 of 2 CDROM.
4. Type the command `mkdir /jumpstartboot`
5. Type the command `cd /cdrom/cdrom0/s0/Solaris_9/Tools`
6. Type the command `./setup_install_server -b /jumpstartboot`

The following messages should appear on the screen

Verifying target directory...

Calculating space required for the installation boot image

Copying Solaris_9 Tools hierarchy...

Copying Install Boot Image hierarchy...

Install Server setup complete

7. Type the command `cd /jumpstartboot/Solaris_9/Tools/Boot`
8. Type the command `ls -l`

The `/jumpstartboot` directory holds all the files that a JumpStart client needs to boot. This is basically a "mini Solaris operating system." Remember, the JumpStart client does not boot from its own operating system. It boots from the JumpStart Boot server's mini Solaris boot image.

9. Type the command `cd /`
10. Type the command `eject cdrom`

Lesson 21.10 Using `add_to_install_server` Script

This lesson copies the contents of the Solaris 2 of 2 Software CDROM onto the JumpStart server's hard drive. The Solaris 9 Languages CDROM also has an `add_to_install_server` script located in the `/s0/Solaris_9/Tools` directory.

1. Log in as the root user.
2. Open a Terminal window.
3. Insert the Solaris 9 Software 2 of 2 CDROM.
4. Type the command `/cdrom/cdrom0/s0/Solaris_9/Tools/add_to_install_server /exportsoftware`
The command `add_to_install_server` adds the software to the `/exportserver` directory.

Creating a JumpStart Directory

The JumpStart configuration files `rules`, `rules.ok`, `sysidcfg` and the profile text files must be in a shared directory. This directory can be anywhere and can have any name. After the JumpStart clients load a miniature version of the Solaris 9 operating system from the JumpStart Installationserver, the clients then read the JumpStart configuration files from the JumpStart Configuration server.

Lesson 21.11 Sharing a JumpStart Directory

In this lesson readers learn how to set up and share a JumpStart directory. This directory resides on the JumpStart Configuration server.

1. Log in as the root user.
2. Open a Terminal window.
3. Edit the `/etc/dfs/dfstab` file with the `vi` editor and make the changes shown below.
 - a. Add the line `share -F nfs -o ro,anon=0 /JumpStart`
 - b. Add the line `share -F nfs -o ro,anon=0 /exportsoftware`
 - c. Add the line `share -F nfs -o ro,anon=0 /jumpstartboot`
4. Type the command `shareall`
The `shareall` command reads the `/etc/dfs/dfstab` file and shares the directories listed in the file.

If the `shareall` command does not work, try typing the command `/etc/init.d/nfs.server start`

Lesson 21.12 Updating The JumpStart Files

The `rules` file is particularly important here, because it needs to be updated with the latest client information. Unlike the `rules` file for a typical Local JumpStart installation which is usually only used for one type of hardware, the `rules` file for a Network JumpStart installation is typically used for different types of hardware and operating system configurations, depending on the network segment the workstation resides on.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command `cd /JumpStart`
4. Copy any desired `rules` file, `myprofile` file and the `sysidcfg` file to this directory (if they do not already exist).
5. Type the command `cp rules rules.backup`
This command creates a backup copy of the `rules` file. If this file becomes too badly damaged, type the command `cp rules.backup rules` to restore the saved version.
6. Type the command `vi rules`

Update the **rules** file with any configuration changes necessary. Read the "Understanding the Rules" section in this chapter for information on the **rules** file.

7. Type the command **vi myprofile**
Edit this file so that it matches the **rules** file. Read the "Understanding the rules File" and "Understanding a profile_text_file" sections in this chapter for information on creating the **rules** file and the profile text files. This **rules** file still needs to be checked with the **check** command.
8. Type the command **check rules**
Make sure that the check was OK, and that a file named **rules.ok** was created by the check utility.
9. Type the command **vi sysidcfg**
Edit the **sysidcfg** file so that it matches the client's settings.

The add_install_client script

This script modifies important directories and files on the JumpStart server. It can be found in the **/cdrom/cdrom0/s0/Solaris_9/Tools** directory.

The **add_install_client** script supports the following options:

add_install_client <options>

<options>

- c <server:/configuration/directory>** Identifies the profile server. This option specifies the location of the configuration directory. The configuration directory contains files such as the **rules** file, the profile files, and the begin and finish scripts.
- e <ethernet address>** Specifies the MAC address (also known as an Ethernet address) of the client. This is not required if the **/etc/ethers** file has the MAC address or if a naming service knows the client's name.
- f** Specifies the client's boot file name.
- i <IP address of client>** Specifies the IP address of the client. This is not required if the IP address of the client is already in the **/etc/inet/hosts** file. This option is not required if a naming service knows the client's IP address.
- p <server:/sysidcfg/directory>** Specifies the directory where the **sysidcfg** file is located.
- s <server:/install/directory>** Tells the client where the Install server is located. It specifies the location where the CDROM images are shared.

The sysidcfg file

A Network JumpStart installation usually requests that a NIS or NIS+ server be present on the network for the JumpStart installation to occur. The **sysidcfg** file can be used when a NIS or NIS+ server is not present on the network. This file must reside in the same directory as the **rules** and **rules.ok** files, and the profile text files. Figure 21.2 shows a typical **sysidcfg** file.

```
system_locale=en_US
timezone=US/Mountain
timeserver=localhost
name_service=none
```

```

network_interface=primary { hostname=SUN100
                             ip_address=192.168.0.4
                             default_route=192.168.0.1
                             netmask=255.255.255.0
                             protocol_ipv6=no }
security_policy=none

```

Figure 21.2 A Typical sysidcfg File

The **sysidcfg** file above contains the following keywords:

system_locale	Same information as from the <code>/usr/lib/locale</code> file.
timezone	Same information found in the <code>/usr/share/lib/zoneinfo</code> file, the timezone such as: mountain, GMT, etc.
timeserver	Defines where the system obtains it's official time.
name_service	The type of name service such as: DNS, LDAP, NIS, NIS+ or none.
network_interface	Settings for the ethernet card(s) on the system.
hostname	The hostname of the system, type the command hostname to see this.
security_policy	Sets the type of security such as Kerberos, none.

Files Used with a Network JumpStart Installation

This section shows some typical JumpStart files. In most cases, you will not need to modify these files by hand.

The `/tftpboot` directory

This directory contains the **inetboot** files for each client. The **inetboot** file works just like the **bootblk** file on a local hard drive.

The **inetboot** file has the following naming convention:

```
inetboot.<hardware_architecture>.OSVersion
```

For example, the Ultra5 Workstation is based on the sun4u architecture. To install Solaris 9 on a Ultra5 workstation there should be a file named.

```
inetboot.sun4u.Solaris_9-1
```

The `/etc/dfs/dfstab` File

This text file is used to share the directory or directories of the JumpStart server. This file holds the directory's share information. Figure 21.3 shows the contents of a typical `/etc/dfs/dfstab` text file.

```

share -F nfs -o ro,anon=0 /exportsoftware
(for systems that have the CDROM image copied to the hard drive)

share -F nfs -o ro,anon=0 /cdrom/sol_9_sparc/s0/Solaris_9/Tools
share -F nfs -o ro,anon=0 /cdrom/sol_9_sparc/s1" to /etc/dfs/dfstab
(for systems that are using the CDROM as the source of the software image)

```

Figure 21.3 The `/etc/dfs/dfstab` File

The /etc/bootparams File

The `/etc/bootparams` file gives the initial boot parameters to a client. The server must be running the `rpc.bootparamd` daemon. Figure 21.4 shows a typical `/etc/bootparams` file.

```
client1 root=sun100:/cdrom/sol_9_sparc/s0/Solaris_9/Tools/Boot install=sun100:/
exportsoftware boottype=:in sysid_config=sun100:/JumpStart install_config=sun100
:/jumpstartboot rootopts=:rsize=3276
```

Figure 21.4. A Typical /etc/bootparams File

If this file does not seem to make a lot of sense right now. It will be easier to understand after you run the `add_install_client` script. For now, just know that the `add_install_client` script creates this file. It is used with workstations that boot off of the network. A client that is performing a Network JumpStart installation is obviously booting off of the network.

The `/etc/bootparams` file contains the following types of information:

myclient	Hostname of the client.
root=	Location of the Boot server. This directory is where the root file system is spooled to.
install=	Location of the spooled software image.
boottype=	Type of Boot server.
sysid_config=	Directory that holds the configuration files for the JumpStart installation.
install_config=	Location of the Install server. This directory contains the operating environment installation files.
rootopts=	Mount options for the root file system.

The /etc/inetd.conf file

This file is used by the `init` process to start a program when a request comes in from a particular TCP port. Figure 21.5 shows a small clip from the file.

```
# TFTP - tftp server (primarily used for booting)
#tftp dgram udp6 wait root /usr/sbin/in.tftpd in.tftpd -s /tft
pboot
```

Figure 21.5 /etc/inetd.conf File Contents

When a TFTP request comes in, the `init` process starts the `in.tftpd` daemon to take bootp requests. Basically this file says “when a network service is requested (telnet, FTP, TFTP, SSH), start the following process (`in.telnetd`, `in.ftpd`, and so forth).

The /etc/nsswitch.conf File

The `/etc/nsswitch.conf` file is used to point to a name server. This name server can be a NIS, NIS+, or LDAP server. The name server should have the hostname and IP address of the client. If the name server does not have the hostname and IP address of the client, the `/etc/hosts` file must contain this information for the Network JumpStart installation to continue. This installation sets up the client’s hostname and IP address in the `/etc/hosts` file.

The `/etc/hosts` File

The `/etc/hosts` text file is used to match a hostname with an IP address. The JumpStart client must have an entry in this file for the Network JumpStart to work properly. Figure 21.6 shows a typical `/etc/hosts` file.

```
100.100.100.103    myserver loghost
100.100.100.104    computer1
```

Figure 21.6 A Typical `/etc/hosts` File

The `/etc/ethers` file

The `/etc/ethers` file is a text file used to match a JumpStart client's hostname with an Ethernet address. The JumpStart client usually does not have an operating system installed. For TCP/IP communications to occur, the client needs an IP address. The `/etc/ethers` file resides on the JumpStart server. Figure 21.7 shows one entry in a typical `/etc/ethers` file, with its MAC address and hostname.

```
8:0:20:3b:4c:3    computer1
```

Figure 21.7 A Typical `/etc/ethers` file

When the JumpStart client first contacts the JumpStart server, the client gives the server its Ethernet address. The JumpStart server then looks in the `/etc/ethers` file for a matching address. When a match is found, the server knows what hostname matches that address.

The server then looks in the `/etc/hosts` file for the hostname it just discovered. The `/etc/hosts` file relates hostnames to their corresponding IP addresses. Now the server tells the client its IP address. If the `/etc/hosts` file does not have a matching hostname for an IP address entry, the JumpStart server contacts a name service for the IP address that matches the hostname. (A name server is a network server, such as DNS, NIS or NIS+, that relates a hostname with an IP address.) If the JumpStart server can not ultimately match the client to its Ethernet address, the message **Timeout waiting for ARP/RARP packet** is displayed on the client's screen.

After the IP address is given to the client, The Boot server directs the client to the location of the JumpStart Installation server and Configuration server. These can be separate servers, or they can be combined in the JumpStart Boot server

Lesson 21.13 Perform the JumpStart Installation

WARNING

This lesson installs a new copy of the Solaris 9 operating system. Always use extreme caution when performing this type of installation. If you have only one hard drive, and something goes wrong, your workstation will not boot properly.

If you have only one hard drive, use EXTREME CAUTION if you follow this lesson!

If you have two hard drives, you should remove the hard drive that has a known good copy of Solaris 9 before you follow this lesson. That way, if something goes wrong, you can still boot your workstation from the existing boot drive.

This is the final lesson in a Network JumpStart installation. In this lesson readers will create a new server or workstation from a JumpStart installation.

In this lesson, the term "JumpStart client" refers to the system onto which the operating system is being installed. The term "JumpStart server" refers to the system that holds the JumpStart files that will be installed on the JumpStart client.

1. Connect the JumpStart client to the same network segment as the JumpStart server.
2. Power on the JumpStart client.
3. If the JumpStart client system is powered off, start the boot process and immediately press the STOP + A keys (Sun keyboard) or CTL + Break keys (PC keyboard). This brings the system to the OK prompt.
4. On the JumpStart client, type the command **banner** at the OK prompt.
*The **banner** command will show the system's Ethernet address, on the third line. It should look something like **8:0:20:3b:4c:3**.*
5. On the JumpStart server, type the command **cd /cdrom/cdrom0/s0/Solaris_9/Tools**
6. On the JumpStart server, type the command
cp add* /exportsoftware/Solaris_9/Tools
7. On the JumpStart server, type the command **share**
*The **share** command shows the directories that have been shared on the network. The **/JumpStart**, **/exportsoftware** and **/jumpstartboot** directories should be shared. If they are not shown, review Lesson 21.11.*
8. On the JumpStart server, type the command **vi /etc/hosts**
9. Add the following line **192.168.0.10 client1**
*This line associates the hostname **client1** with the IP address **192.168.0.10** for the JumpStart installation.*
10. On the JumpStart server, type the command **cd /exportsoftware/Solaris_9/Tools**
11. On the JumpStart server, type the command **./add_install_client -e 8:0:20:3b:4c:3 -i 192.168.0.10 -c sun100:/jumpstartboot -p sun100:/JumpStart -s sun100:/exportsoftware client1 sun4m**

In this example, the client with the hostname **client1** and the **sun4m** architecture was set up with the IP address **192.168.0.10**. The Ethernet address of the client should be **8:0:20:3b:4c:3** (as shown by the **banner** command at the **OK** prompt).

The client architecture can be found with the command **uname -a**. The output of this command is shown below:

```
./add_install_client -e 8:0:20:3b:4c:3 -i 192.168.0.10 -c sun100:/jumpstartboot
-p sun100:/JumpStart -s sun100:/exportsoftware client1 sun4m
Adding Ethernet number for client1 to /etc/ethers
saving original /etc/dfs/dfstab in /etc/dfs/dfstab.orig
Adding "share -F nfs -o ro,anon=0 /exportsoftware" to /etc/dfs/dfstab
making /tftpboot
enabling tftp in /etc/inetd.conf
starting rarpd
starting bootparamd
updating /etc/bootparams
copying inetboot to /tftpboot
```

12. On the JumpStart client, type the command **boot network - install**

13. Sit back and watch the show.

Do not be too surprised if the JumpStart installation does not work properly the first time. It is a very tricky process to get a JumpStart installation to work properly

If the installation did not succeed and you were installing to a second hard drive on the JumpStart client, no harm has been done. Simply run the installation again..

Some Final Comments on the Networked JumpStart Installation

As you have just seen, the Networked JumpStart installation process is *very complex*. The following discussion summarizes it. It's not unusual to be a little unclear about the process, even after you have been through it once. But experience is a good teacher. As you gain experience with the process, anything that is a bit unclear now will start to become clearer.

Networked JumpStart

The JumpStart installation can occur from local media or from a network connection. The Solaris 9 operating system can be installed locally or from a network. There are three types of servers that are required for a JumpStart installation. These services can be provided by one server or by three different servers.

- **Boot and client identification services** (Boot server). This server has the boot information for the JumpStart client to boot up. Remember, the JumpStart client does not typically have an operating system installed. For the installation to take place, the client needs to load an initial operating system from the Boot server. This is done over a network connection.
- **Installation services** (Install server). This contains a copy of the Solaris CDRoms. The Install server can share its CDRom drive, or it can have the contents of the CDRom on its hard drive and share that directory. Table 21.1 shows the required CDRom images for the installation of different types of clusters. To install a Developer or Entire Distribution of Solaris 9, the CDRom images must be spooled to the hard drive.
- **Configuration services** (Configuration server). This service provides configuration information for the client. Typical configuration information includes details such as the size of the partitions, the software to load, and install scripts. The configuration services can be custom tailored to the type of server being constructed. For example, it's possible to specify that E-250 servers should have FTP service installed, while Ultra 5 workstations should not have the FTP service installed.

Key Points to Remember

Solaris 9 supports several different types of installations. This chapter demonstrated a JumpStart installation on a single workstation. Most companies use the JumpStart installations over the network. The average SPARC server does not contain a diskette drive, so a Local JumpStart is not a viable option. There are only slight differences between a Local JumpStart installation and a Network JumpStart. A JumpStart installation can be difficult, especially for someone who is not used to the process. Do not be discouraged if the first attempt to create a networked JumpStart fails.

Chapter 22 Role Based Access Control

Lessons in This Chapter

Lesson 22.1 Creating Rights in SMC	22-8
Lesson 22.2 Using an Administrative Role	22-11
Lesson 22.3 Creating Rights in SMC	22-11
Lesson 22.4 Making a Custom Role by Hand	22-16

Introduction

With previous versions of Sun Solaris there were only two types of accounts: the root user and an ordinary user. The root user was an all-powerful account that could do everything, and the ordinary user was very limited in what commands and functions were available. On one hand, this was a good system, because the generic user really could not damage the server. On the other hand, the system administrator was typically the only person with root access to the system and was very busy performing simple tasks. Power users were unable to perform some high end tasks they needed for their jobs. Power users had to make a request to the system administrator to perform high end tasks on their behalf.

One solution was to use an open source tool known as **sudo**. This application was created to let ordinary users have some of the rights of a system administrator. The only drawback was that **sudo** assigns system administrator rights directly to a user, which makes accountability an issue.

Solaris 8 introduced a new concept, RBAC (Role Based Access Control). With RBAC a user logs in as himself or herself. After a proper login, users could assume a "role" by using the **su** command. While users are in this role, their accounts are temporarily disabled. This improves system security. Each role has its own administrative shell, such as **psh** (Profile Bourne Shell), **pfcs** (Profile C shell), and **pfksh** (Profile Korn shell). It is still possible to assign administrative rights to a user directly, but this is not a good idea, due to security concerns.

RBAC is used to give ordinary users the ability to perform some root level functions. For example, only the root user can format a hard drive. Under RBAC it is possible to give trusted users the right to format a hard drive. In some ways this is a double-edged sword. Users have more power to do more things that reduce the system administrator's workload, but the same users now have the ability to damage the server. Most system administrators are extremely careful when they work on a server. Unfortunately, novice users may not be equally careful. The real problem comes with the user who wants to "learn new things" and use a running production server as a learning tool. This chapter will demonstrate how to create rights and roles, both by using the SMC (Solaris Management Console), and by hand.

Key RBAC Concepts

Unfortunately, the Sun Solaris RBAC is a very difficult thing to master at this time. On the back end, it presents a confusing variety of data fields, attributes, key words, delimiters, rights and rules to the system administrator. There are times when it appears that a role has been created, but a user still can not perform the tasks that this role is designed to perform. To understand RBAC, go slowly. Learn the basic concepts first, then slowly add more RBAC skills.

The most common use of RBAC is to give a junior system administrator an account with limited rights. This is a person who is not trusted with the root account. Another common use of RBAC is to give lower-level help desk personnel a specific role that they have been trained to perform. Resetting passwords is a good example of a role that a help desk person can perform and not crash a server.

Understanding Rights

A rights can be loosely thought of as an “administrative function.” This lets a user do something that is ordinarily reserved for the root user. A right can be assigned to a role, or it can be given directly to a user account.

There are some security and auditing rights that should never be given to an ordinary user. An ordinary user could disable a security feature to perform a task and then not reset the security after the task is complete. There is also the possibility a user could misuse the security rights for internal hacking.

Some of the key rights in Solaris 9 are as follows:

All	The user has the ability to run any command inside the administrator’s shell. This only lets the user run commands associated with his or her normal account. If a user is given Print Manager and All permissions, that user still can not format a hard drive.
Audit Control	The user has the ability to manage the audit subsystem, but not to read audit files. Most system administrators would not give this to a typical user due to security concerns (it could be used to cover one’s tracks). Audit managers and security experts should be given this right.
Audit Review	The user can read audit files but can not change what is audited. This is not a harmful thing to give to a manager or security expert.
Cron Management	The user has control over the crond daemon and the cron table, including other users’ cron tables. This role could cause some problems, so it should only be given to someone with prior system administration experience.
DHCP Management	The user has the ability to manage the DHCP services. This is a very critical function that can damage client connections. This role should only be given to someone with system administration experience. A novice user should never ever have this right.
Device Management	This right grants the user the ability to work on a device, add a device or remove a device from the system. Only an experienced power user should have this right.
Device Security	The user can configure a device, but not add or remove the device. Only someone inside the system administration staff should have this right.
FTP Management	This right lets the user work with the FTP server. This is a good right to give to a Web developer.
File System Management	The user has the ability to perform mounts and shares. This is a reasonable right to give a typical user.
File System Security	This gives a user the ability to manage the security on a file system. Only security experts and junior system administrators should be granted this right. Ordinary users don’t need this at all.

Log Management	This right grants the user the ability to manage log files. Only security experts should use this, because it could be used to cover one's tracks in case of a security concern.
Mail Management	The user can manage <code>sendmail</code> . Only people directly associated with email should use this. Users don't need this for their email systems.
Maintenance and Repair	The user can maintain the system. This is a job for the root user; it should not be delegated away.
Media Backup	The user can create backups on a backup device (tape devices). This is a very good right to grant to even novice users. Just make sure they don't take the concept too far, like creating a full backup every hour.
Media Restore	This is another good right to give a user. Just make sure the user knows how to make sure that files are only restored to his or her own directory.
Name Service Management	This lets the user manage the name service daemon. Name service is something like LDAP, NIS, NIS+. Don't grant this right to anyone outside the system administration staff.
Network Management	This gives a user the ability to affect the network settings on a server. Only experienced system administrators should have this ability. In general, avoid delegating this role.
Network Security	This role should only be given to security experts. Ordinary users do not have any reason to work with trusted network databases. This could be abused by someone trying to do something by disabling security. This potential gap in security could be exploited by a hacker.
Object Access Management	This right lets the user change file ownership and permissions. This is a very reasonable right to grant to ordinary users.
Printer Management	This is often given to users. Printers tend not to be mission critical devices so even if a user screws up a printer, it's not a critical issue.
Process Management	Don't delegate this right. Only experienced system administrators should work with processes. The root user already has the ability to work with processes.
Project Management	This right gives the user the ability to manage projects. Software Team Leads and Project Managers can be given this right.
Rights Delegation	Only give this right to junior system administrators. The Rights Delegation function lets users delegate only the same rights they own themselves.
Software Installation	This is a good right to grant to trusted users. A user can add and remove software to the system. Just don't let the users manage system critical software, such as Sun Management Center or DHCP.
SunScreen	SunScreen is the built-in firewall for Solaris 9. This right should not be given to anyone except network engineers. If a mistake is made with the firewall, a server can become inaccessible or become insecure from a security standpoint.

User Management

This is a good right to give to a junior administrator. This user can create and modify users. This does not include the ability to change user passwords.

User Security

This is a great right to give to a junior system administrator or help desk support person. This role can create or modify a user's password.

Understanding Roles

A role can be thought of as a special user account that has rights assigned to it. A role has a login name, a UID (User IDentification) number, and a home directory. A role works in a special shell known as an *administrative shell*. A user can not just log in as a role, because that would cause a security problem.. Instead, a user must log in with an ordinary user account and then assume a role with the **su** command. When a user assumes a role, the user's account is disabled while they are inside the role.

In the past, there were only two types of accounts, the root user and a regular user. As mentioned above, a role is a specially created user account that can run some administrative commands. For example, a system administrator could create a user account named **myformatuser** and then give this account the right to use the **format** command. Roles are defined in the **/etc/security/prop_attr** database.

As mentioned above, for security purposes, users can not log in directly to a role account. Try to log out and then log back in under a role account. It is impossible.

The only way to use a role account is to log in as an ordinary user then enter the **su** command, along with the username for that role account. For the special "format-enabled" account mentioned above, this would be

```
su myformatuser
```

Remember that the **su** command is used to transform one user into another user. The reason the role user can not log in directly is for system accountability. If someone logged in as **myformatuser** and reformatted a hard drive, you would want to know who did it! This is easy, because the **su** command keeps a log file. If John Smith damages a server, the **su** log will show that **johnsmith** was the user who used the **su** command to become myformatuser and then damaged the server.

Understanding Authorization

An authorization grants permission to a restricted command. Ordinarily, a restricted command is a command that only the root user can run. Authorizations allow other users, roles or profiles to run a restricted command. Authorizations are defined in the **/etc/security/auth_attr** database file.

The key point is that a role is a special user account. An authorization is granted to the role. An ordinary user needs to log in as himself or herself and then assume the role with the **su** command.

Understanding Profiles

Profiles are used to assign the same authorizations to a large number of users. A profile is nothing more than a collection tool. A profile can not do anything in and of itself. A profile collects authorizations, users and roles together to assign rights to many users at the same time. Profiles are defined in the **/etc/security/prop_attr** database file.

Profiles run in special shells, These shells are:

psh Profile Bourne Shell

pfcs	Profile C Shell
pfksh	Profile Korn Shell

Understanding Execution Attributes

Execution Attributes define the commands that can be executed by the users and roles assigned to a profile. Execution Attributes are defined in the `/etc/security/exec_attr` database file.

Commands Used with Roles

The commands listed below allow system administrators to manage roles. **Some of these** commands create, modify, or delete a role. Other commands show what authorizations and profiles are associated with a role.

roleadd This command creates roles. Most of the options used with **roleadd** are the same options used with the **useradd** command. The **roleadd** command supports the following options:

roleadd < options >

- A** Assign an authorization to the role.
- c** Create a comment.
- d** Home directory of the role.
- m** Make the home directory if it does not exist.
- p** Assign a profile to the role.

rolemod This command modifies an existing role. The **rolemod** command supports the following options:

rolemod <options>

- A** Assigns comma separated authorizations to the role.
- e** Sets an expiration date for the role.
- l** Creates a new loginID.
- p** Assigns comma separated profiles to the role.
- s** Assigns a new shell to the role.

Other related commands include:

roledel Deletes a role (deletes the user account that is the role).

auths Shows what authorizations are associated with a user.

profiles Shows what profiles are associated with a user.

roles Shows what roles are associated with a user.

Databases Associated with RBAC

In the `/etc` directory there are four text files that are used with RBAC. These text files serve as the collection point for a simple database. The databases can be broken down into two logical groups.

The first group can be called RBAC database files. These are text files used in the background. They are used to create and define elements such as roles, authorizations and profiles. These text files are very complex, but it is crucial that a system administrator understand how they work, just in case RBAC does not perform as it is supposed to and the files need to be manipulated by hand.

The first logical group defines profiles and attributes:

`/etc/security/exec_attr`

This database defines profiles and their privileged operations.

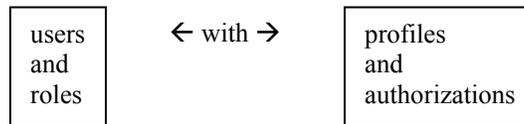
`/etc/security/auth_attr`

This database defines authorizations and their attributes (also includes related help files).

The second logical group associates elements of RBAC with each other:

`/etc/user_attr`

This database associates four elements together:



`/etc/security/prof_attr`

This database associates profiles with authorizations (also includes related help files).

The remainder of this chapter covers the RBAC text files in detail. It is recommended that readers try to understand these files in a general sense. There are times a system administrator might have to modify one of these files by hand, as shown in the next lesson.

The `user_attr` database file

The `user_attr` database has the following fields:

user:qualifier:res1:res2:attr

- user** The loginID for this username in the `/etc/passwd` file.
- qualifier** Reserved for future use.
- res1** Reserved for future use.
- res2** Reserved for future use.
- attr** This field has the following sub-fields:
- auths** Authorizations associated with a user.

profiles	Profiles associated with a user.				
roles	Roles associated with a user.				
type	This field has the following sub-fields:				
	<table> <tr> <td>normal</td> <td>Normal user.</td> </tr> <tr> <td>role</td> <td>Role user.</td> </tr> </table>	normal	Normal user.	role	Role user.
normal	Normal user.				
role	Role user.				

The **auth_attr** database file

The **/etc/security/auth_attr** database has the following fields:

authname:res1:res2:short_desc:long_desc:attr

authname This is the only field that has any real value. It is the name of the authorization, in the format of <prefix.suffix.perm>. The other fields are just used to describe the authname field.

The **authname** field has the following three subfields:

prefix	A descriptive tag that describes what is making the authorization. This can be the keyword <i>Solaris</i> (for Sun OS authorizations) or the reverse DNS name of a domain, such as <i>com.microsoft</i> , <i>com.uswest</i> or <i>com.novell</i> .				
suffix	What is being authorized, such as <i>printmgr</i> or <i>adminsuite</i> .				
perm	This field has the two sub-fields. They are: <table> <tr> <td>grant</td> <td>Can grant other users the same permissions.</td> </tr> <tr> <td>admin</td> <td>Has administration rights.</td> </tr> </table>	grant	Can grant other users the same permissions.	admin	Has administration rights.
grant	Can grant other users the same permissions.				
admin	Has administration rights.				

res1 Reserved for future use.

res2 Reserved for future use.

short_desc Short text description.

long_desc Long text description.

attr Keywords separated by semicolons (;) that describe the attributes of an authorization. The keyword "help" references an HTML file that contains help information on the authorization, such as *help=MyAuthorHelp.html*.

The **prof_attr** database file

The **/etc/security/prof_attr** database has the following fields:

prof:res1:res2:desc:attr

prof Name of the profile (case sensitive).

res1 Reserved for future use.

Res2	Reserved for future use.
desc	Long text description.
attr	This field has the following sub-fields:
help	A help file associated with this profile. For example, MyProf2.HTML.
auths	Authorizations from the auth_attr database file.

The **exec_attr** database file

The **/etc/security/exec_attr** database has the following fields:

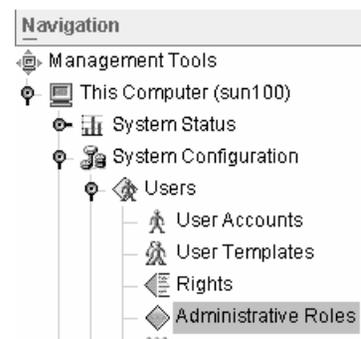
name	Name of the profile (case sensitive).
policy	Only valid policy entry is <i>suser</i> .
type	Only valid type is <i>cmd</i> .
res1	Reserved for future use.
res2	Reserved for future use.
id	The actual command (such as /usr/sbin/format) with the full path name.
attr	This field has the following sub-fields:
eu id	Username or UID.
u id	Username or UID.
eg id	Effective GID.
g id	Group ID.

Lesson 22.1 Creating Rights in SMC

In this lesson a custom-made right will be created with the SMC tool. These rights will later be assigned to an administrative role, and then a user will be added to the role. This section only shows how to create a right in the SMC.

Note: At any time, you can back up a screen by left clicking the < Back button or cancel by pressing the Cancel button.

1. Log in as the root user.
Open a Terminal window.
To open a Terminal window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Tools menu item, then left click on the Terminal icon.
2. To start the Sun Management Center at the command line, type the command **smc&**
The SMC icon is also on the Workspace menu. To start the SMC from the workspace menu, right click anywhere in unoccupied desktop space, left click on Tools icon, then left click on the Solaris Management Console icon.



3. Left click on the "This Computer (sun100)" icon.
 4. Left click on the "System Configuration." icon.
 5. Left click on the "Users" icon
- If a popup window appears with the title "Log in: User Name" log in as the root user.*
6. Left click on the "Administrative Roles" icon.
 7. Left click on the Action menu bar item.
 8. Left click on the Add Administrative Role... menu choice.

The first window creates the name and description of the role.

9. Create a role with the name **softadder**
 - Fill in the fields as follows:
 - a. Role Name: softadder
 - b. Full Name: softadder
 - c. Description: Role to add software
 - d. Role ID Number: Will be chosen automatically. Do not change.
 - e. Role Shell: Administrator's Bourne should be selected.
 - f. Create a role mailing list: uncheck this check box.

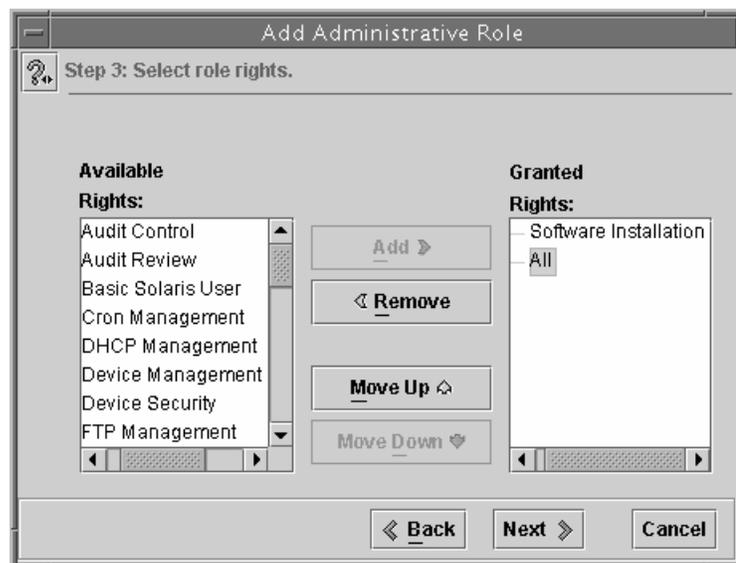
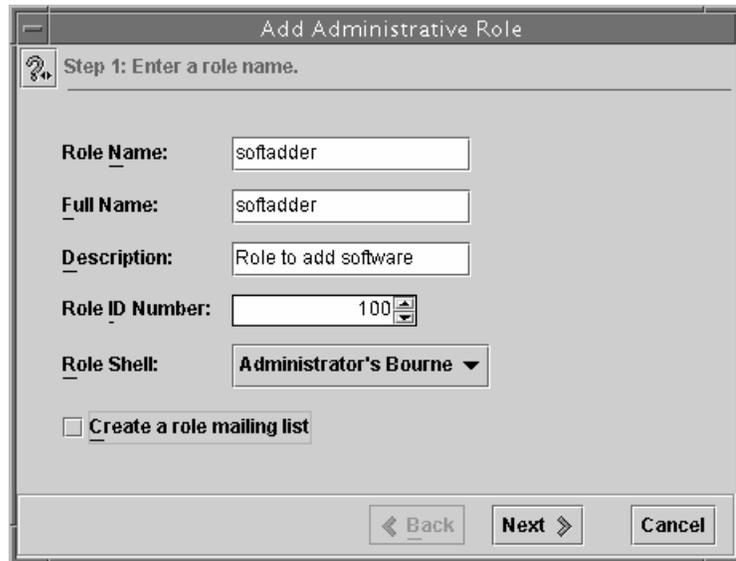
The next window asks the for the administrative role's password.

10. Type in **user22** for the password.

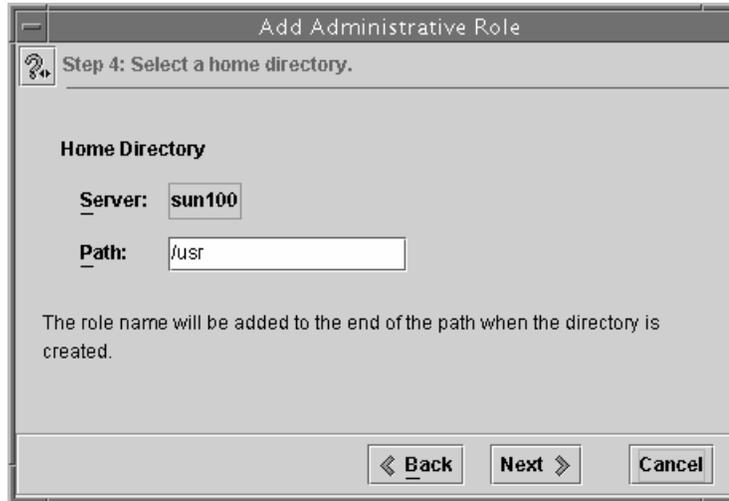
The next window lets the administrator choose what rights should be assigned to the administrative role.

11. In the Available Rights field (at the left), scroll down and select Software Installation. Then left click on the Add > button to move this option to the Granted Rights window (at the right). This adds only the right that has been selected.
12. Select All, left click on the Add > button to move this option to the right window. This adds *all rights* from the Available Rights window
13. Left click on the Next > button.

The next window prompts for the home directory of this role.



14. Type in `/usr`

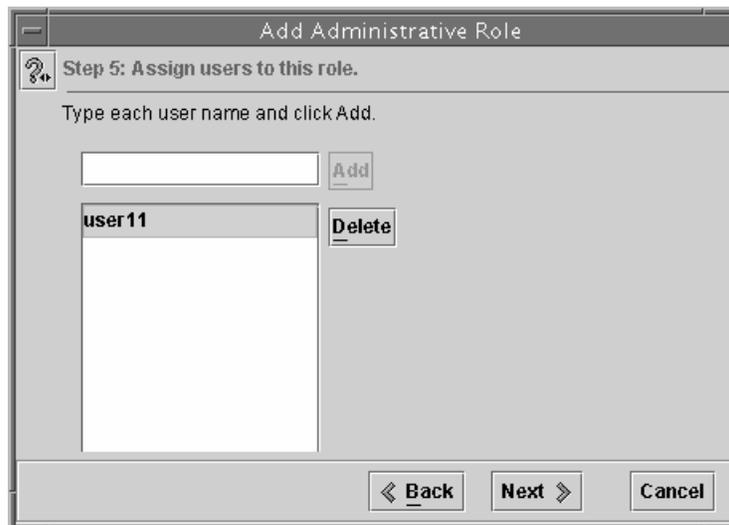


15. Left click on the Next > button. The next window asks for a user name to assign to the role.

16. Type in `user11`

If user11 does not exist, create user11. Creating users is covered in Chapter 7.

17. Left click on the Next > button. The next (and final) screen is a configuration screen.



18. Left click on the Finish button



Lesson 22.2 Using an Administrative Role

In this lesson the reader will use user11 to create another user. User11 will log in to the system, then **su** to the **softadder** role. After user11 has created another user, user11 will quit the **softadder** role.

1. Log in as user11.
2. Open a Terminal window.
3. Type the command **su softadder**

*Special note: Do not use the command **su - softadder**. The dash (-) character with **su** imports the environmental variables of the target new user. This causes problems with Administrative Roles. The password for softadder is user22 as typed in above.*

4. Type the command **id**
*The account **user11** should now be **softadder**.*
5. Type the command **/usr/sbin/pkgadd**

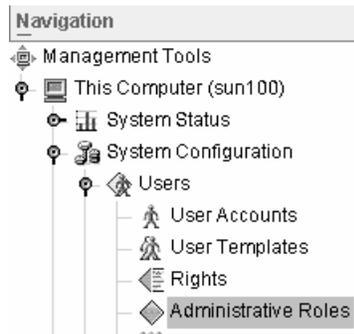
*Only the root user can add software with the **pkgadd** command. In this case, the role **softadder** can also do so. The error message, **pkgadd: ERROR: no packages were found in </var/spool/pkg>** happens because the **pkgadd** command was only used to illustrate RBAC, and the user is not giving the proper commands to actually install a software package.*

Lesson 22.3 Creating Rights in SMC

In this lesson readers modify a user by giving that user the right to create and modify other users. The special user can also change the password of other users.

Note: In the following lesson, after filling in fields shown in a window, click the Next > or OK button to continue. You can also go back a step by clicking the < Back button, or cancel by clicking the Cancel button.

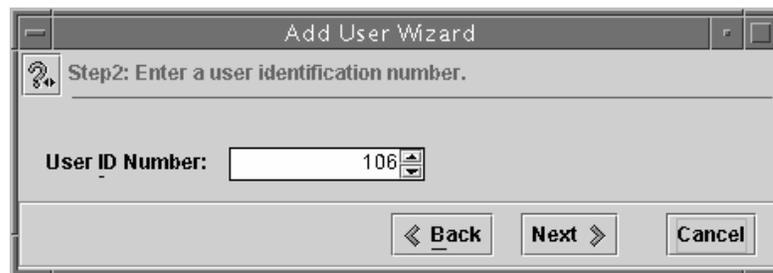
1. Log in as the root user.
2. Open a Terminal window.
3. Type the command **smc&**
The SMC icon is also on the Workspace menu. To start the SMC from the Workspace menu, right click anywhere in unoccupied desktop space, left click on Tools icon, then left click on the Solaris Management Console icon.
4. Left click on the “This Computer (sun100)” icon
5. Left click on the “System Configuration.” icon
6. Left click on the “User Accounts” icon
7. Left click on the Action menu bar item.
8. Left click on Add User
9. Left click on With Wizard...
10. In the first window, shown below, type in this information:



User Name : testuser
Full Name : Test user
Description : RBAC Test User



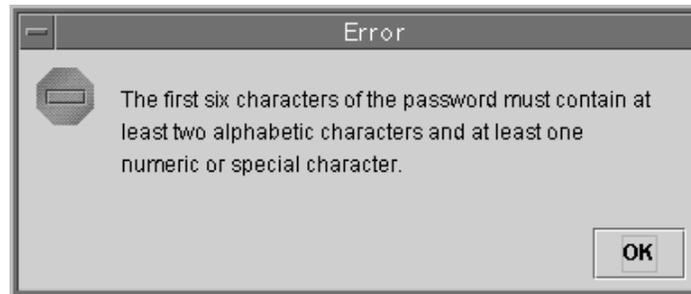
11. In the second window, shown below, enter this information:
User ID Number: Let the system pick this,



12. In the third window, enter this information:
User Must Use This Password At First Login: select this choice
Password: test2user
Confirm Password: test2user



If a password is typed that does not match the security rules for Solaris ,9 a warning popup window will appear.



13. In the fourth window, choose this information:
Primary Group : staff



14. In the fifth window, type in this information:
Path: /usr



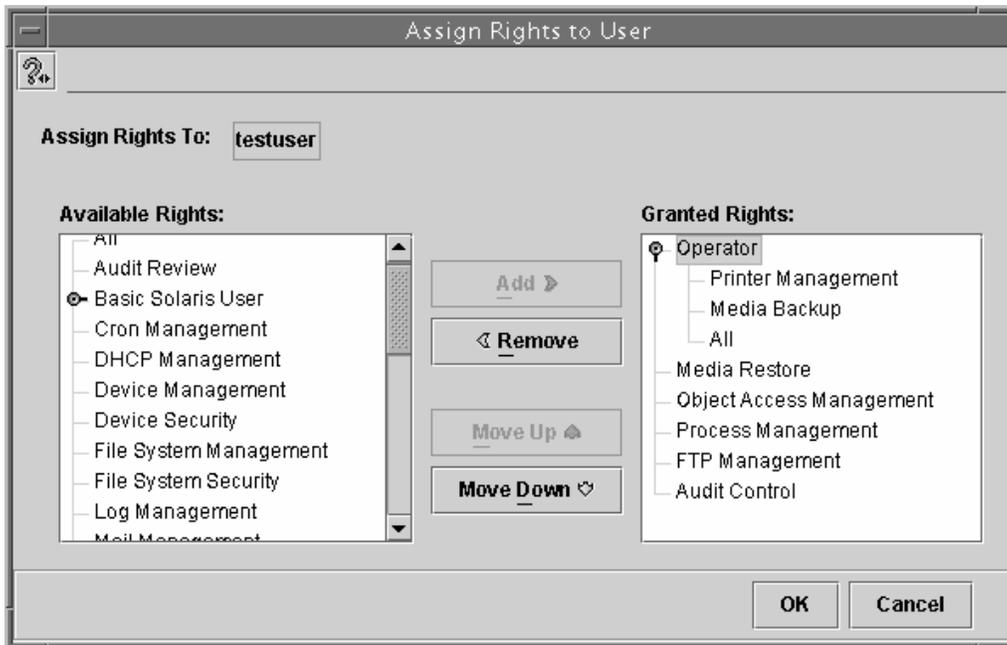
The sixth window is just a confirmation window for the Mail server.



15. In the seventh window, left-click on the Finish button to confirm the settings.



16. To view the user, left click on View menu bar item.
17. Highlight the **testuser** account
18. Left click on the Refresh menu bar item.
19. Left click on the Action menu bar item.
20. Left click on Assign Rights to User...



21. Left click on the OK button

Lesson 22.4 Making a Custom Role by Hand

This lesson shows how to create a role by directly editing the RBAC text files. This lesson creates a profile that will let ordinary users run the **snoop** command.

1. Log in as the root user.
2. Open a Terminal window
3. Type the command **cp /etc/user_attr /etc/user_attr.backup**
This backs up the /etc/user_attr file in case of error. The next three steps perform similar functions.
4. Type the command
cp /etc/security/prof_attr /etc/security/prof_attr.backup
5. Type the command
cp /etc/security/exec_attr /etc/security/exec_attr.backup
6. Type the command
cp /etc/security/auth_attr /etc/security/auth_attr.backup
7. With a text editor, modify the **/etc/security/exec_attr** file. Add this line at the end of the file
Snooper:user:cmd::/usr/sbin/snoop:uid=0
*This line allows the profile called **Snooper** to use the **snoop** command.*
8. With a text editor, modify the **/etc/security/prof_attr** file. Add dd this line at the end.
Snooper:::User can use snoop:
9. Type the command
roleadd -m -P "Snooper,All" -d /usr/snprole snprole
*The account **snoprole** will be added to the /etc/passwd file with the **pfsh** (Profile Bourne Shell).*
10. Type the command **passwd snprole**
11. At the **New Password:** prompt, type **snprole1**
12. At the **Re-enter new Password:** prompt, type **snprole1**
13. Type the command **useradd -m -d /usr/user101 -R snprole user101**
*Now, **user101** can **su** to **snprole** and run the **snoop** command.*
14. Type the command **passwd user101**
15. Change the password to **user101**
16. Log out.
17. Log in as user101
18. Open a Terminal window
19. Type the command **/usr/sbin/snoop**
The screen should show the message:
snoop: /dev/eri: Permission denied
*Why would it do that? Haven't we just assigned user101 to the role **snprole**? The answer is that user101 has not used the command **su snprole**. Understand that user101 is just an ordinary user and thus has no right to run the **snoop** command(same as every other ordinary user on the system).*
20. Type the command **id**
The screen should show something that looks like this:
uid=113(user101) gid=1(other)
21. Type the command **su snprole**
*When asked for the password, type **snprole1***
22. Type the command **id**
Notice the UID and username.
23. Type the command **/usr/sbin/snoop**
*Now that user101 has used **su** to take the role of **snprole**, the command works.*
24. Press **CTL+ C** simultaneously to quit the **snoop** command.

Key Points to Remember

A role is a very useful in that it allows an ordinary user to perform tasks that in the past only the system administrator could perform. Remember, this is a double-edged sword. On one hand, it is nice to let the end user take care of himself or herself. The problem comes with the fact that some users are not careful and could damage the server. Try to make custom roles by hand. These roles will only have the specific commands required for the required task. If a role is selected through the SMC, there might be some commands in the role that are not safe for an ordinary user to use.

Chapter 23 NIS and NIS+

Lessons in This Chapter

Lesson 23.1 Preparing the NIS Server to Use an Alternate Source Directory.....	23-6
Lesson 23.2 Creating a NIS Master Server.....	23-8
Lesson 23.3 Create a NIS Client.....	23-9
Lesson 23.4 Working with NIS.....	23-10
Lesson 23.5 Remove NIS from Clients and Servers.....	23-11
Lesson 23.6 Set Up a NIS+ Server.....	23-18
Lesson 23.7 Working with NIS+ Tables.....	23-21
Lesson 23.8 Removing NIS+.....	23-22

Introduction

Quick Tip

- Most of the lessons in this chapter can be followed if you have only one computer. However, Lesson 23.3, Lesson 23.4 and Lesson 23.6 require one computer to be the server, and a second computer to be the client.
- If you only have one Sun system, you may be able to set up a client computer if you have an Intel-based system that is running either Solaris 8, or a system that is running a recent version of Linux. See your system's documentation as necessary.

One very common problem that large companies face is the difficulty of updating a large number of servers with a limited staff. For example, if a company had 400 SPARC servers and needed to add this line to the `/etc/hosts` file:

```
23.145.23.207    data32
```

it would take several days for one person to do this. There would also be a high possibility of error making this change 400 times.

NIS (Network Information Server) allows system administrators to change text files in a configuration directory on the NIS Master server (usually `/etc`) and then have the NIS clients automatically update their text files to match the NIS Master server. Under this scenario, the system administrator can update the `/etc/hosts` file on the NIS Master server and then push it to all the NIS clients. Now all the computers in the company would have an updated `/etc/hosts` file that matches the NIS Master server.

Each domain can have only one NIS Master server. If this server becomes overwhelmed with NIS requests, NIS slave servers can be created. NIS Slave servers just handle NIS requests from clients. Understand that an NIS Slave can not update NIS information.

This type of computing is known as a client-server architecture. NIS was created to provide a centralized administration of different types of network services. The information that NIS stores is known as the *NIS namespace*. This namespace typically includes text files located in the `/etc` directory. A NIS domain is a flat hierarchy, in that there are no other levels of NIS servers other than NIS Master server, NIS Slave servers and NIS client servers.

As mentioned above, each NIS domain provides centralized administration. Unfortunately, this can be a problem for very large organizations, because they can have only one NIS domain server. Usually NIS has to be broken down into different domains and structures. So most organizations set up multiple NIS domains. This tends to become very cumbersome, because each NIS domain must be administered separately.

Another drawback is that all NIS map updates are done as a whole. NIS maps are groups of text files that centralize the storage of important data from configuration files in the `/etc` directory. Finding information in a NIS map file is much quicker than looking for it in `/etc` files. These files will be covered in greater detail later in this chapter. Unfortunately, there is no such thing as an incremental NIS map update. All the information is updated or not updated. Any time there is a change, the NIS maps have to be created again and then propagated to the NIS clients.

NIS+ was created to overcome some of the limitations of NIS. One of the most important features is that NIS+ domains can be hierarchical in nature. Instead of only having one domain, **CompanyX**, there can be several domains, such as **Sale.CompanyX**, **Engineering.CompanyX**, and **Marketing.CompanyX**. Additional subdomains can also be used. For example **EuropeX** can be created with the subdomains **Sales.EuropeX.CompanyX**, **Engineering.EuropeX.CompanyX** and **Marketing.EuropeX.CompanyX**. These domains can have separate administration or they can have central administration.

NIS and NIS+ update their information in different ways. NIS+ uses tables and can perform what is known as an *incremental update* to the table. In NIS+, changes propagate almost instantly. A NIS server requires a new map file be created and propagated to all NIS clients and NIS Slave servers whenever even a single piece of information changes. With NIS, it takes time to create a map and update all NIS clients.

So, why would anyone want to use NIS instead of NIS+? The main reason is that NIS is a lot easier to work with than NIS+. Even if a network has only 10 workstations, setting up NIS+ can be complicated. The only drawback for a small company in using NIS is that it is not as secure as NIS+. NIS can not be used in a highly secure environment.

Note: Most of the NIS lessons in this chapter can be followed if you have only one computer. However, Lesson 23.3, Lesson 23.4 and Lesson 23.6 require one computer to be the server, and a second computer to be the client.

If you only have one Sun system, you may be able to set up a client computer. For the client, you will need either an Intel-based system running Solaris 8, or any system that is running a recent version of Linux. See your system's documentation as necessary.

Common NIS Terms

Before a full discussion of NIS can occur, the reader needs to understand some common NIS terms.

NIS Master Server	This server contains the master copies of the NIS files. The NIS Master server makes copies of the <code>/etc</code> text files into NIS maps. A NIS map file is easier for Solaris 9 NIS to read and work with than a text file. The NIS Master Server is the central point of authority for the NIS system.
NIS Slave Server	This server contains backup copies of the NIS files. The NIS Slave server lightens the load of the NIS Master Server by taking NIS client requests. If the NIS Master Server

went down, the NIS Slave Server would still take NIS client requests. The NIS Slave can only keep a copy of the NIS Master's NIS tables. Most network engineers believe that thirty NIS clients can be serviced by a single NIS Slave. Under light loading conditions fifty NIS clients can be serviced by a single NIS Slave.

NIS Client

These are the workstations and other servers that receive information from a NIS Master or NIS Slave server. Their `/etc` text files look just like the NIS master or NIS client. They have no authority in the NIS domain. They do not have any NIS maps internally. If a NIS Master Server is unavailable it will bind to a NIS Slave Server to update its `/etc` files. If there is a NIS bind failure the NIS client will look for another NIS Slave to work with

NIS Maps

NIS servers like to work with NIS maps. NIS maps provide a direct map to files in the `/etc` directory. Each NIS map has two columns of information. The first column is the primary key for the information, and the second column is an information key for the first column.

Example : the `/etc/hosts` file has two primary pieces of information:

Hostname	IP address
<code>data32server</code>	<code>213.14.32.103</code>

To map this information, NIS creates two files:

`hosts.byname` -Maps hostnames and addresses, using the *hostname* as the information key.

`hosts.byaddr` -Maps hostnames and addresses, using the *address* as the information key.

How NIS Map Files Are Named

NIS map files are located in the `/var/yp/<NIS domainname>` directory. NIS maps are named after the file they represent. For example, one NIS map file is:

map.key.pag

map Name of the NIS map, usually matches the file name from the `/etc` directory. For example, `hosts` from `/etc/hosts`.

key NIS uses a database concept known as a *sort key*. The sort key is used by the system to look up the data more quickly.

For example, a file could list the IP addresses for hosts in the first column (the sort key) and the host names in the second column (the data). This might look something like this:

192.168.0.9	nisboy
127.0.0.1	localhost
192.168.0.4	sun100

If an `/etc` text file has several columns of data, a separate key file is created for each column. The key name is something like `byhosts`, `byIP`, `bydate`, and so forth.

pag The NIS maps data

For example, the following map files exist for the files `/etc/hosts` and `/etc/group`:

Map	Key	Pag	Map File Name
hosts	byaddr	pag	hosts.byaddr.pag
hosts	byname	pag	hosts.byname.pag
group	bygid	pag	group.bygid.pag
group	byname	pag	group.byname.pag

If the data file (the file that ends in **pag**) is very large, an indexing file is created. This indexing file helps the NIS system find information faster.

The index file follows the format

map.key.dir

dir For very large NIS maps an index file is used to sort the NIS maps.

For example, the following index files exist for the files **/etc/hosts** and **/etc/group**:

Map	Key	Dir	Index File Name
hosts	byaddr	dir	hosts.byaddr.dir
hosts	byname	dir	hosts.byname.dir
group	bygid	dir	group.bygid.dir
group	byname	dir	group.byname.dir

The files for the NIS Master server are in the **/var/yp** directory. Some common NIS files are shown in Table 23.1.

NIS Common Names	Map Name	What the NIS Map Does
bootparams.root , bootparams.swap	Bootparams.*	Maps /etc/bootparams (contains path names that diskless clients need when they start)
ethers.byaddr , ethers.byname	ethers.*	Maps /etc/ethers (correlates NIC addresses with IP addresses)
group.byid group.byname	Group.*	Maps /etc/group (lists group names and group IDs)
hosts.byaddr hosts.byname	Hosts.*	Maps /etc/hosts (correlates hostnames to IP addresses)
mail.byuser mail.byaddr	mail.*	Maps mail files in the /etc directory
netgroup.byuser netgroup.byhost	netgroup.*	Maps /etc/netgroups (lists network groups by host or user)
netmask.byaddr	netmask.*	Maps /etc/netmask (lists IP

NIS Common Names	Map Name	What the NIS Map Does
		addresses and netmasks)
networks.byaddr networks.byaddr	networks.*	Maps /etc/networks (lists network names and addresses)
passwd.byuser passwd.byuid	passwd.*	Maps /etc/passwd (lists user information such as UID, shell, and so forth)
protocols.byname protocols.bynumber	protocols.*	Maps /etc/protocols (lists the protocols understood by the NIS Master).
rpc.bynumber rpc.byservice	rpc.*	Maps /etc/rpc (lists rpc port numbers)
services.byname services.byservice	services.*	Maps /etc/services (supplies protocols and ports)

Table 23.1 Common NIS Maps

When files in the **/etc** directory on the NIS Master are updated, the system administrator *can* modify the file **/var/yp/Makefile** to reflect these changes. This is covered below.

When files in the **/etc** directory on the NIS Master are updated, two commands *must* be run to update the information on NIS slaves and NIS clients.

These commands are:

cd /var/yp Switches to the directory that has the makefile.

/usr/ccs/bin/make Updates the NIS maps by reading the makefile in the **/var/yp** directory.

Common NIS Commands

The NIS server, NIS slaves and NIS clients all use these common NIS commands. All NIS commands start with the two letters **yp** (NIS+ commands all start with three letters **nis**. The commands shown below are NIS commands):

ypmake Updates the NIS maps by reading the make file in the **/var/yp** directory.

ypcat Works just like the **cat** command for regular files. This command writes out all the text from a NIS map.

ypinit	One of the primary commands used to create NIS Masters, NIS Slaves and NIS Clients. This command builds the NIS database. It also updates the lists on NIS clients and servers.
ypoll	Gets the polling number from a NIS Master or NIS Slave. The poll number indicates how old the NIS maps are.
ypush	When the NIS maps are updated on the NIS Master, the NIS Slaves must receive the updates. The ypush command pushes out the new NIS maps to the NIS slaves. It does not push the NIS maps out to clients.
ypset	Used on a client to bind that client to a particular NIS Slave or NIS Master. This is very useful if load balancing issues start to occur.
ypstart	Starts the NIS service after a NIS Master or NIS Slave has been created. This command is also used if a NIS Master or NIS Slave dies out for some reason.
ypstop	Stops all NIS services on a master, slave or client.
ypwhich	This command is very useful for troubleshooting NIS problems. The ypwhich command returns the name of the NIS Master or NIS Client that supplied a NIS map. This is good for connectivity issues and for troubleshooting map corruption.

Lesson 23.1 Preparing the NIS Server to Use an Alternate Source Directory

As noted above, by default, a NIS server is set up to replicate files from the **/etc** directory located on the NIS Master server. Unfortunately, this is not a good idea, because the **/etc** directory contains the **/etc/passwd** and **/etc/shadow** files for the root user. A hacker who gains access to a NIS client can then pull down the **passwd** and **shadow** files from the NIS Master. This can provide the hacker with useful information, such as the shell the root user is using, and if another root user has been created. Understand that it is possible to create a user account and then change its UID to zero (0) which makes an ordinary user into a root user.

One solution to this problem is to set up the NIS server to use an alternate source directory (other than **/etc**). The following lesson shows how to prepare the system for this.

1. Log in as the root user.
2. Open a Terminal window.
To open a Terminal window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Tools menu item, then left click on the Terminal window icon.
3. Type the command **cd /var/yp**
4. Type the command **cp Makefile Makefile.backup**
*This command copies **Makefile** to **Makefile.backup**. If something goes terribly wrong, type the command **cp Makefile.backup Makefile** to restore the **Makefile** back to it's original condition.*
5. Type the command **vi Makefile**
Change the line
DIR =/etc
to
DIR=/mynis/etc

If IPv6 is installed change
INETDIR=/etc/inet
to
INETDIR=/mynis/etc/inet

If RBAC will be used, change
RBACDIR=/etc/security
to
RBACDIR=/mynis/etc/security

Also, change
PWDIR=/etc
to
PWDIR=/mynis/etc

6. Save the **Makefile** and quit the **vi** editor
7. Type the command **mkdir /mynis**
*This creates the directory **/mynis** that will be used to hold the alternate **etc** directory*
8. Type the command **mkdir /mynis/etc**
*This creates the **/mynis/etc** directory, where the original **/etc** files will be copied.*
9. Type the command **mkdir /mynis/etc/security**
*This creates the **/mynis/etc/security** directory, where several security files will be copied.*
10. Type the command **mkdir /mynis/etc/inet**
*This creates the **/mynis/etc/inet** directory, where the **ipnodes** file will be copied. This is only necessary if IPv6 is installed.*
11. Type the command **cd /etc**
12. Type the following commands.
*Each command below copies the file listed after **cp** to the directory listed at the end of the command. If you get an error message when you try to copy a file (in this step or any other step in this lesson), use the **touch** command to create it, and then try to copy the file again. For example, if the **/etc/ethers** file does not exist, type the command **touch /etc/ethers** and then type the command **cp /etc/ethers /mynis/etc**. These two steps create a blank file and then copy that file to the **/mynis/etc** directory.*

```
cp /etc/auto_home /mynis/etc
cp /etc/auto_master /mynis/etc
cp /etc/bootparams /mynis/etc
cp /etc/ethers /mynis/etc
cp /etc/group /mynis/etc
cp /etc/hosts /mynis/etc
cp /etc/ipnodes /mynis/etc
cp /etc/netgroup /mynis/etc
cp /etc/netmasks /mynis/etc
cp /etc/networks /mynis/etc
cp /etc/passwd /mynis/etc
cp /etc/protocols /mynis/etc
cp /etc/rpc /mynis/etc
cp /etc/services /mynis/etc
cp /etc/shadow /mynis/etc
cp /etc/user_attr /mynis/etc
cp /etc/timezone /mynis/etc
cp /etc/publickey /mynis/etc
```
13. Remove any line that has the root user, or any user with **UID=0**.
14. Save the file.
15. Type the command **vi /mynis/etc/shadow**
16. Remove any line that has the root user or any user with **UID=0**
17. Save the file.
18. Type the command **cp /etc/inet/ipnodes /mynis/etc/inet**
19. Type the command **cp /etc/security/audit_user /mynis/etc/security**
20. Type the command **cp /etc/security/auth_attr /mynis/etc/security**
21. Type the command **cp /etc/security/exec_attr /mynis/etc/security**

22. Type the command `cp /etc/security/prof_attr /myNIS/etc/security`

Lesson 23.2 Creating a NIS Master Server

Now that an alternative source directory has been set up, you can create a NIS Master server. In this case, this will be your workstation. The NIS Master server will contain the original NIS Master map files and will have authority over the NIS domain. This server is created with the `ypinit -m` command.

Note: You must perform the previous lesson before performing this one.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command `cp /etc/nsswitch.conf /etc/nsswitch.mybackup`
This command copies the original `nsswitch.conf` file so that the original file is still on the client if it needs to be used again.
4. Type the command `cp /etc/nsswitch.files /etc/nsswitch.conf`
This command updates the `/etc/nsswitch.conf` file so that the server reads its information from its own `/etc` files. This is crucial for a NIS server.
5. Type the command `/usr/sbin/ypinit -m`
A message like the following will appear after the command is typed:

In order for NIS to operate successfully, we have to construct a list of the NIS servers. Please continue to add the names for YP servers in order of preference, one per line. When you are done with the list, type a <control D> or a return on a line by itself.

next host to add: sun100
next host to add:

6. Press CTL+D simultaneously. This exits from the “**host to add:**” questions from the `ypinit` command.
The next message will be similar to:
The current list of yp servers looks like this:
sun100 ← *This will be the hostname of your system*
Is this correct? [y/n] ← *answer y*
7. *If the following questions appear, answer **y**:*
 - **Do you want this procedure to quit on non-fatal errors?**
This should not be a problem with a fresh install of Solaris 9
 - **Can we destroy the existing /var/yp/aassddff.com and its contents?**
These files only exist if the server was once a NIS Master before. These are older out of date files that should be replaced.
8. At this point, the NIS server should be installed successfully. If so, congratulate yourself and continue with the next step!

If the NIS server did not install correctly the first time, don't be surprised! Just read the error messages. They should give you the information you need to fix the problem. If the installation fails, it is usually because a file was not copied over correctly.

After correcting any problems, run the command `/usr/sbin/ypinit -m` (step 5) again. Each time this command is run, it completely destroys the previous information and creates a new NIS server. This ensures that partially bad data is not injected into the new NIS server.

9. When a message like **sun100 has been setup as a yp master server without any errors** appears, type the following command:

```
/usr/lib/netsvc/yp/ypstart
```

The screen should display the message

```
starting NIS (YP server) services:  
ypserv ypbind  
ypxfrd rpc.  
yppasswdd rpc.  
ypupdated done.
```

10. Type the command **ypcat passwd**

*This command works just like the **cat** command. It displays all the information from the NIS server on a particular file.*

A NIS Master server has now been created on the server. However, other computers on the network will not honor this NIS Master until they have been set up as NIS clients. The next section shows how to set up the NIS clients.

The NIS Client

Solaris 9 can function as a NIS Master and as a NIS client. The only steps necessary for creating a client are to set up a domain on the client and change settings in the **/etc/nsswitch.conf** file to look for a NIS server, instead of for the workstation's local files.

Lesson 23.3 Create a NIS Client

In this lesson Solaris 9 will set up as a NIS client. This is generally an undramatic process.

Note: The following lesson requires *two* Solaris workstations on the network. The NIS *server* must be set up on a Solaris 9 workstation. If you only have one Solaris workstation, you can set up the NIS client on an Intel-based machine that is running Solaris 8, or on a machine that is running most recent versions of Linux.

1. On the NIS client, log in as the root user (all steps in this lesson are done on the client).
2. Open a Terminal window.
3. Type the command **domainname**

If a domain name has been set up on this machine, this command will echo that name. If this happens, go to the next numbered step.

If a domain name has not been set up, nothing will be shown. If this happens, follow these two substeps:

- a. Type the command **domainname aassddff.com** (or a domain name of your choice).
*This command sets the domain name to **aassddff.com** (or whatever you typed) on the system. In the following steps, substitute your domain name if you did not enter **aassddff.com**.*
 - b. Type the command **domainname**
Make sure that the domain name is correct. If not, go the previous substep.
4. Type the command **echo "aassddff.com" > /etc/defaultdomain**
*The file **/etc/defaultdomain** sets the domain name automatically when the system reboots. If this file does not exist, the domain name of the system will be blank when the system reboots.*
 5. Type the command **cat /etc/defaultdomain**
This command displays the domain name of the system.
 6. Type the command **vi /etc/hosts**
Make sure the NIS server's hostname and IP address are in this file and are correct.
 7. Type the command **ping -s <NIS_host_name>**.
For example:
ping -s sun100

The **ping** command makes sure that the NIS server's hostname can be contacted. If the hostname is valid, the IP address and hostname should send back ICMP ping reply packets.

8. Type the command **/usr/sbin/ypinit -c**

A message appears that reads:

In order for NIS to operate successfully, we have to construct a list of the NIS servers. Please continue to add the names for YP servers in order of preference, one per line. When you are done with the list, type a <control D>

or a return on a line by itself. next host to add: 192.168.0.4 ← Enter your NIS Master IP address or hostname

next host to add: (press CTL + D)

In this example, **192.168.0.4** is the IP address of the NIS Master server. You can enter either the hostname or IP address. As shown above, press CTL + D at the second **next host to add:** prompt.

The final message should be

The current list of yp servers looks like this:

192.168.0.4

Is this correct? [y/n: y] type y

9. Type the command **/usr/lib/netsvc/yp/ypstart**

This command starts the **yp** daemons on the client.

10. Type the command **/usr/lib/netsvc/yp/ypbind -broadcast**

This command starts the **ypbind** daemon. The **ypbind** daemon needs to be running in order for NIS to work.

11. Type the command **ypwhich**

This should return the name of the NIS server that is responding to this NIS client.

12. Type the command **cp /etc/nsswitch.conf /etc/nsswitch.backup**

This makes a backup copy of the **/etc/nsswitch.conf** file. If something goes terribly wrong, type the command **cp /etc/nsswitch.backup /etc/nsswitch.conf** to restore the **nsswitch.conf** file to its original condition.

13. Type the command **cp /etc/nsswitch.nis nsswitch.conf**

This overwrites the **/etc/nsswitch.conf** file with the **/etc/nsswitch.nis** file. The **/etc/nsswitch.nis** file tells the client to obtain its information from the NIS server.

Lesson 23.4 Working with NIS

In the previous two lessons the NIS server and a NIS client were set up. This lesson shows how to change NIS tables on the NIS Master server and how to configure the client for NIS services.

Note: The following lesson requires *two* Solaris workstations on the network. The NIS *server* must be set up on a Solaris 9 workstation. If you only have one Solaris workstation, you can set up the NIS client on an Intel-based machine that is running Solaris 8, or on a machine that is running a recent version of Linux.

On the NIS Master Server:

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command **vi /mynis/etc/group**

The **group** file should look something like this:

4. Add this line to the end of the file:

doggroup::2000: ← IP address of the NIS server

(Make sure there is not an existing group with a GID=2000).

The line **doggroup::2000:** creates a new group in the NIS source file **/mynis/etc/group**. This updated entry is not active in the NIS directory until the following two commands are typed.

5. Type the command **cd /var/yp**
6. Type the command **/usr/ccs/bin/make**

The **make** command uses a text file known as the **Makefile**, which directs the **make** command to compile new NIS maps. Type the command **man make** to read more information on the **make** command. It is used with a lot of programs.

A message like
updated group
pushed group
updated netid
pushed netid
should be displayed.

On the NIS Client:

7. Type the command **ypcat group | grep dog**

The results of this command should look like this:

doggroup::2000:

The point of this lesson was to make a rather harmless NIS entry that would not damage the NIS client or server if a mistake occurred. The group **doggroup** can not be used for anything. It is just a harmless entry in the NIS tables.

How to Remove NIS

There are times when a NIS server and NIS clients need to be removed from a server or workstation. This section gives the general steps to remove a NIS server from Solaris 9. Lesson 23.5 shows the necessary steps in detail.

The first step is to remove NIS from the clients. If the NIS server is removed first, there will be lost NIS clients that will be looking for the server.

1. To remove NIS from a client, the first step is to kill the NIS processes. Almost all NIS commands and processes start with **yp** as the first two letters.
2. After the NIS processes are killed, restore the original **nsswitch.conf** file back to its original condition before NIS was set up on the client. In previous lessons, the file **nsswich.backup** was created to save the original **nsswitch.conf** file.
3. The last step is to remove the **/var/yp/bind** directory (**cd /var/yp** then type **rm -rf bind**) and reboot the client.

Removing NIS from the server follows almost the exact same steps, except that it is a good idea to save the NIS information in a compressed file, such as a **tar** or **zip** or **gz** file.

Lesson 23.5 Remove NIS from Clients and Servers

This lesson first removes NIS from a client system. After the clients have NIS removed, the NIS server can be disabled.

Note: If you did not install any NIS clients, start with the **On the Server** section, below.

On the NIS Client:

1. Log in as the root user.
2. Open a Terminal window

3. Type the command **ps -ef | grep <process name>**
*The command **ps -ef | grep <process name>** is used to find a **Process ID #**, then the command **kill <process ID # >***
4. Type the command **ps -ef | grep <process name>**
5. Type the command **cp /etc/nsswitch.mybackup /etc/nsswitch.conf**
*If a backup file was not made of the **nsswitch.conf** file, type the command **cp /etc/nsswitch.files /etc/nsswitch.conf***
6. Type the command **cd /var/yp**
7. Type the command **rm -rf binding**
8. Type the command **ps -ef | grep <process name>**
9. Type the command **ps -ef | grep <process name>**
10. Type the command **ps -ef | grep <process name>**
11. Type the command **ps -ef | grep <process name>**
12. Type the command **ps -ef | grep <process name>**
The next step is optional, but it is generally a good idea.
13. Type the command **init 6**
*The command **init 6** reboots the client system. This is a good idea because the client might still have some residual programs that are still trying to contact a NIS server.*

On the NIS Server:

14. Log in as the root user.
15. Open a Terminal window.
16. Type the command **ps -ef | grep <process>**
*The command **ps -ef | grep <process>** is used to find a **PID** then **kill PID#***
17. Type the command **ps -ef | grep <process>**
18. Type the command **ps -ef | grep <process>**
19. Type the command **ps -ef | grep <process>**
20. Type the command **ps -ef | grep <process>**
21. Type the command **cd /var/yp**
22. Type the command **tar -cvf backupnis.tar ***
*This saves the contents of the **/var/yp/** directory in the file **backupnis.tar***
23. Type the command **tar -uvf backupnis.tar /mynis**
*Copies the contents of the **/mynis** directory into the tar file.*
24. Type the command **tar -uvf backupnis.tar /etc/hosts**
*Copies the contents of the file **/etc/hosts** into the tar file.*
25. Type the command **tar -tvf backupnis.tar | more**
*Check the output of this command. Make sure the NIS domain information was saved to the **backupnis.tar** file. If there is a problem, go back to step22.*
26. Type the command **rm -rf aassddff.com** (if your domain has a different name, use that name instead).
*This deletes the domain directory **aassddff.com** from the **/var/yp** directory*
27. Type the command **rm *time**
28. Type the command **rm -rf /mynis**
29. Type the command **vi /etc/hosts**
*Edit the **hosts** file for any IP addresses that are no longer needed, such as the IP addresses of NIS clients or NIS Slave servers that do not need to be contacted. These hostnames and IP addresses are not harmful if they are left in the file. Just remember to update your **/etc/hosts** file if the hostnames of these systems change in the future.*
30. Type the command **rm -rf binding**

The binding directory might not exist. If so, ignore the error message.

31. Type the command **cp Makefile.backup Makefile**
*This restores the original **Makefile** to its default condition, provided that the **Makefile.backup** was created when the NIS server was created.*
32. If possible, reboot the server.
Some companies are very sensitive about rebooting a server. so it might be a good idea to check the scheduled downtime on a server.

NIS+

NIS+ can be thought of as a more advanced version of NIS. One key point is that NIS+ is not just a remake of NIS. It is a completely new software program that performs the same functions as NIS but is much more advanced. The NIS+ domain consists of the *NIS+ name space*. The domain is defined by the root directory, the **org_dir** directory and the **groups_dir** directory.

One of the key differences between NIS and NIS+ is that NIS+ can be set up in a hierarchical manner. With NIS, a company with the domain name **adams.com** could only have one domain. Any NIS servers had to accommodate all the clients in that single domain. But NIS+ allows for a hierarchy of domains. For example, the company could now have multiple domains such as **sales.adams.com**, **marketing.adams.com** and **production.adams.com**.

NIS+ also allows each domain to be administered on a local level. The sales department could have a network engineer or system administrator set up the NIS+ master and NIS+ replica servers for that domain. The production department could hire a system administrator to set up and maintain a NIS+ master server just for the **production.adams.com** domain. The only catch is that there has to be a root Master Server for the entire **adams.com** domain.

Another key difference between NIS+ and NIS is that NIS+ uses what are known as *tables* instead of NIS maps. A NIS+ table can hold more than two entries (NIS maps only hold two entries). Because the tables hold more entries, there is no need to have multiple maps for a namespace object, as there is in NIS. Tables also feature indexes that allow for faster searches. A NIS+ server's tables can be created from NIS maps or from files in the **/etc** directory. This allows for quicker NIS-to-NIS+ upgrades.

NIS+ also offers increased security. With NIS any client that requested NIS information received that information. This presented a security hole because a hacker could simply join a network through a weak client and then pretend to be a NIS client. The NIS server would unquestioningly send down NIS information that the hacker could use without being authorized.

Understand that one of the key things a hacker does is to obtain any information possible. This information may be critical information (a root user's password) or something mundane (another client's IP address.) These pieces of information can be linked together for further hacking attempts. Basically, a hacker needs to learn as much as possible about the target network. A NIS server that dumps that information without questioning the client's authenticity makes this easier.

With NIS+ clients need to have a NIS+ password to set themselves up as a NIS+ client. NIS+ uses what is known as DES2 security. This makes snooping the setup information more difficult. Basically, a NIS+ server looks for two key pieces of information: authorization and authentication. Authorization answers the question "Who is authorized to look at this information?" and Authentication answers the question "How do I know you are who you claim to be?"

A NIS+ server can be set up in what is known as NIS+ compatibility mode. In this mode the NIS+ server fields requests from both NIS and NIS+ clients. This allows for relatively seamless migration from NIS to NIS+. After the NIS+ server is set up, the NIS clients can be slowly and carefully converted over to NIS+ clients. Once all

the clients have become NIS+ clients, NIS+ compatibility mode can be turned off for security's sake. From then on, any request from a NIS client will come in as unauthenticated NIS request.

On the negative side, NIS+ is a lot more complex to setup than NIS. Some of the complexity comes from new features, such as dealing with security issues, setting up tables, and setting up client authorizations. This means that if a network is fairly secure behind a firewall and the client data security is not critical (such as for workstations in a classroom) it is much easier to just set up NIS. On the other hand, environments that require the maximum amount of network security should use NIS instead of NIS+.

NIS+ Tables

NIS+ stores its information in tables, instead of in the map files created by NIS. These tables are more efficient than map files. On the down side, they are very complex to update and view, compared to NIS maps.

There are 16 primary NIS+ tables. Additional tables can be created in NIS+ for any purpose imaginable. Understand that in NIS+ a name service replaces the text files that are ordinarily found in the `/etc` directory.

The sixteen primary NIS+ tables are (in no particular order):

1. **hosts.org_dir**
2. **netgroup.org_dir**
3. **services.org_dir**
4. **bootparams**
5. **mail aliases**
6. **protocols**
7. **passwd**
8. **timezone**
9. **RPC**
10. **cred**
11. **networks**
12. **netmasks**
13. **auto_home**
14. **group**
15. **ethers**
16. **auto_Master**

These tables are protected by a security structure that features authorization and authentication. Authorization defines what information will be given to what user. Authentication checks that the requester (the NIS+ principal) is who it really claims to be. NIS uses LOCAL and DES authentication.

As noted above, NIS+ tables replace NIS maps. Data can be retrieved from these tables by using column, entry or table queries. Each domain has its own tables. For example, if a company had the three subdomains **sales.adams.com**, **marketing.adams.com** and **production.adams.com** each subdomain would have its own tables. These tables can share their information with any authorized and authenticated NIS client that makes a request.

NIS+ Name Space

The NIS+ name space holds three types of objects in a domain: directory objects, NIS table objects and group objects. These three primary objects are held in NIS+ directories. The directories can be organized into a domain. A domain logically resembles a company. For example, let's say that the ACME.com company had three main divisions: Production, Sales and Marketing. A NIS+ domain structure could be setup like Figure 23.1.

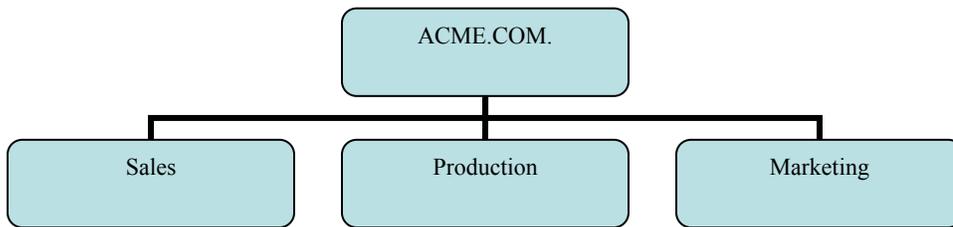


Figure 23.1 Acme.com Domain

A NIS+ domain can be customized to fit an organization's structure. If a new office called *Sales* is created, a NIS+ namespace can be created for that new department.

There are three common types of objects in a NIS+ name space. They are: directories, groups and tables.

Directory Objects A NIS+ directory object works just like a file system directory. It holds other objects and other NIS+ directories.

org_dir This directory holds NIS+ tables.

groups_dir This directory holds groups.

The name for a typical directory structure might look like *sales.usa.acme.com*. The directory names are separated by periods (.) and are read from left to right.

Group Objects These objects store the *NIS+ principals*, which means the users and workstations that are making NIS requests. These objects are used for NIS+ access control.

Table Objects There is a matching NIS+ table for every NIS object. Table objects store server information, such as login names, host names and IP addresses. Each table has its own schema that defines the columns inside the table.

A domain, such as *acme.com*, is serviced by the *Root Master NIS+ server* and *Root Master Replica NIS+ servers*. The Root Master server has authority over the entire domain. Root Replica servers only handle NIS+ requests for the *acme.com* domain. The Root Replica servers only have a copy of the Root Master's tables. They can not change the tables.

Let's try a simple example. Let's say that Acme decided that it needed to use a NIS+ server to update the */etc* text files on all of its Solaris 9 workstations and servers. It buys a SPARC server to be the Solaris 9 NIS+ Root Master server. Solaris 9 is loaded on this server (named *master1*) and the Root Master NIS+ server is created for the *acme.com* domain. This Root Master server is responsible for all the clients in the company under the *acme.com* domain. All the Solaris 9 servers and workstations are converted into NIS+ clients.

As the company starts to grow, the network engineer complains that the Root Master server is too busy to take all the NIS+ requests from NIS+clients. Another Solaris 9 server is set up to be the Root Replica server for the *acme.com* domain. The Root Replica server accepts clients' requests for the *acme.com* domain. This is just like the Root Master server, except it can not alter the NIS+ tables. A replica server can only take NIS+ requests and copy the tables from the master table.

Eventually, Acme grows so large that it decides to create three sub-domains: *sales.acme.com*, *marketing.acme.com* and *production.acme.com*. A new server is purchased for each sub-domain. So, each of the three domains now has its own Master server. NIS+ clients in the *sales.acme.com* domain ask the *sales.acme.com*

Master server for NIS information. The same is true for the NIS+ clients in marketing and production. The each contact their domain's Master server.

So, what happened to the original Root Master server (master 1) ? It is still alive. But instead of taking requests directly from NIS+ clients in all three domains, it feeds information down to the sales, marketing and production master servers. If a new server is added, and it needs to be available to all NIS+ clients, here is all that needs to be done:

- The new server's IP address is set up in the NIS+ table in the Root Master server.
- The sales, marketing and production Master servers then update part of their NIS+ tables from the Root Master server.

When a new server is added to a subdomain, it should be connected *only* to the Master server for its subdomain. Let's say a new CAD (Computer Aided Design) server is added to the production department. This new server should get its information only from the production department's NIS+ Master server. The reader might ask the question, "Would it be better to set up the new CAD server to be able to contact any NIS+ server available, in case the production server goes down?"

The answer to this question is simple: *no*. The NIS+ Master server for each subdomain contains some settings that apply *only to its own subdomain*. If a NIS+ client in the sales subdomain asked for NIS+ information from the production NIS+ Master server, information might be incorrect for clients in the sales subdomain. For example, if the production subdomain used Pacific Standard Time and the sales subdomain used Eastern Standard Time, any time zone information received by a NIS+ client in the sales subdomain from the production NIS+ server would be incorrect.

If a NIS+ Master server becomes swamped with NIS+ requests, it is a good idea to add a NIS+ Replica server to handle the extra NIS+ requests. For example, let's say that after awhile the sales department becomes so large that its NIS+ Master Server is swamped with NIS+ requests. To solve this, the sales department could add a NIS+ Replica server. This Replica server will receive table updates from the Sales Master server and handle some NIS+ requests from clients, lessening the load on the Sales Master server.

NIS+ Table Updates

One of the major improvements of NIS+ over NIS is that the information in its tables is only incrementally updated. By default, these incremental updates only occur every two minutes. This two minute interval helps to make sure that the NIS+ Master server and the NIS+ Replica server don't become too busy making many small changes all the time. The changes are collected every two minutes and then updated in a batch job.

The data that the Master Server sends to a Replica server is sent with timestamps. If a Replica server is down for five minutes and comes back online, it will synchronize itself with the updates from the last five minutes. When a Replica server comes back after thirty minutes it will synchronize itself with the data from the last thirty minutes. Once the Master server realizes that all of its Replica servers have updated their records, it deletes the updates.

If a new Replica server comes online, the Master server will resynchronize *all tables* with the new Replica server. Unfortunately, this is a very intensive effort. While they are synchronizing, the Master server and the Replica server will not even try to answer client requests. The message **NIS+ server is busy, try again** is sent in reply to all NIS+ attempts.

If a network has several Replica servers, this is not a big concern. The Master server and the new Replica server will synchronize all the tables, while the other Replica servers handle the NIS+ requests. But if a network only has one NIS+ Master server, and its first Replica server is coming online, this could cause NIS+ problems with the clients. The best idea is to add the first NIS+ client onto a NIS+ domain at night, or at another time when NIS+ requests are at a minimum.

There are times when a NIS+ Replica server has bad information, and all of its tables need to be re-synced with the Master server as soon as possible. This can be done, but understand that there will be a great deal of network traffic, and that the Master server and Replica server will temporarily ignore client requests, just as when a first Replica server is added to the domain.

Overview of the NIS+ Commands

All NIS+ commands start with the three letters **nis**. These commands are used to create and manage a NIS+ server.

- nissaddent** This command creates the NIS+ tables, from files in the **/etc** directory, from alternate NIS+ source files, or from NIS maps (if upgrading from NIS). This command is only run after the tables have been created and the NIS+ server is set up.
- Nissaddcred** This command creates the NIS+ security credentials for NIS+ principals. It stores information in **cred.org_dir** in the NIS+ domain. The command maps the authentication service name to the NIS+ principal's name and provides authentication information to NIS+ services.
- nisclient** This command initializes the NIS+ credentials for its NIS+ principal. It also creates the initial NIS+ credentials for users and hosts. It restores the network services environment and initializes the NIS+ users and hosts.
- nisinit** This command initializes a server or workstation to be a NIS+ Root Master server or to be a NIS+ client.
- nisserver** This command is used to set up the NIS+ server. It sets up the initial NIS+ directories, such as **groups_dir**. This command can be used to set up the Root Master, Master and Replica NIS+ servers (NIS+ Replica server = NIS+ Slave server). This uses level 2 DES security.
- nisauthconf** This command is run before **nisserver**. This command supports DES, dh640-0 and dh1024-0 authentication. The authentication will proceed with the first authentication mechanism selected. If this command is run as **nisauthconf -v** it will display the NIS+ authentication in use.

The nismaster Command

The **nismaster** command is used to set up a NIS+ Root Master server, a Non-Root master and a NIS+ Replica server.

The **nismaster** command supports the following options.

nismaster <options>

<options>

-d <NIS+ Domain Name>

Sets the NIS+ domain. By default the NIS+ domain is the domain of the server.

-f Forces the setup; no final prompt question will be asked.

-g <NIS+ Group Name>

Specifies the NIS+ group name. Can not be used with the **-R** option.

- h** **<hostname-of-root-master>**
Used to set up Non-Root Masters and Replica servers. This option requests the hostname of the NIS+ Root Master.
- l****<NIS+ password>**
NIS+ password.
- M** Creates the NIS+ server as a Master server.
- R** Creates the NIS+ server as a Replica server.
- r** Sets up the NIS+ server as the Root Master server.
- v** Displays verbose output.
- x** Dry run, displays the commands that will run but does not run them.
- y** Sets up the NIS+ server to be backward compatible with NIS.

Lesson 23.6 Setup a NIS+ Server

Note: The following lesson requires *two* Solaris workstations on the network. The NIS+ *server* must be set up on a Solaris 9 workstation. If you only have one Solaris workstation, you can set up the NIS client on an Intel-based machine that is running Solaris 8, or on a machine that is running most recent versions of Linux.

In this lesson readers are going to perform a dry run method of installing the NIS+ server. The **nissserver** script will only display the output of creating a server. If the dry run method does not produce any error messages, the actual setup of the NIS+server will be performed.

This lesson sets up the **sun100** server on the **aassddff.com** domain to be the NIS+ Root Master server. One big difference between a NIS server and a NIS+server is that once a NIS+ server is created, it needs to be populated with data. After **sun100.aassddff.com** is set up as the NIS+ Root Master server, it will be populated with data from the **/etc** directory. This is done with the **nispopulate** command.

After the NIS+ server is set up, the NIS+ client is set up. This setup procedure involves copying the **/etc/nsswitch.nisplus** file over the **/etc/nsswitch.conf** file. The **nsswich.nisplus** file is a template for the NIS+ services on the client. The **/etc/hosts** file also needs to be modified so that the NIS+ client recognizes the NIS+ server's IP address and hostname.

The final part of this lesson involves creating a fictitious host **192.168.0.5** **dogboy** on the NIS+ Root Master server and adding it to the NIS+ tables. The client then uses the command **ping dogboy** to show that it is getting its information from the NIS+ Root Master server, and not from its own **/etc** text files.

On the NIS+ Server:

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command **/usr/lib/nis/nissserver -x -r -d aassddff.com**
When used with the -x option, the nissserver command operates in dry run mode. Nothing is actually being done, and no files are being copied or modified. The -r option sets up the server as a NIS+ Root Master server. The -v option displays the output in verbose mode. Finally, the -d option sets the NIS+ domain.
 - When asked **"Is this information correct? (type 'y' to accept, 'n' to change) y"** ← Type **y** and then press the Return key.

- When asked “Do you want to continue? (type ‘y’ to accept, ‘n’ to exit this script)” ← Type **y** and then press the Return key.
4. Check for error messages.
 - If no error messages appear, continue with the next step.
 - If an error message does appear:
 - a. Check the `/etc/defaultdomain` file and make sure it has the same domain name as what you specified.
 - b. Type the command `domainname aassdff.com`
 - c. Reinstall the following software packages: **SUNWnisr** and **SUNWnisu**.
To do this, insert the Solaris Software 1 of 2 CDROM (or DVD) and give the following three commands:


```
cd /cdrom/cdrom0/Solaris_9_sparc/s0/Solaris_9/Product
pkgadd -d . SUNWnisr
pkgadd -d . SUNWnisu
```

 Answer **yes** to any prompts that appear.
 - d. Return to step 3 and try the dry run installation again.
 5. If no error messages appear, type the command `/usr/lib/nis/nisserver -r -d aassdff.com -l nis123` (all on one line).
*Because the **-x** option is not being used this time, this is an actual install of the NIS+ server.*
 6. When prompted for the credential for the `sun100.aassdff.com` domain, type the root user’s password. (For an actual production setup, most companies would demand that another password be used.)
Take special note that a group is being created named `admin.aassdff.com`. The principal `sun100.aassdff.com` is added to the `admin.aassdff.com` group.
 7. Type the command `cd /etc`
*The **nispopulate** command must be run in the `/etc` directory (or in the source directory for the file being accessed by this command).*
 8. Type the command `/usr/lib/nis/nispopulate -F -d aassdff.com`
 9. Type the command `/usr/lib/nis/nisping -C aassdff.com`
This command checkpoints the new domain. This makes sure that all servers will support domain transfers to the new domain.

On the NIS+ Client:

10. Log in as the root user.
11. Open a Terminal window.
12. Type the command `cp /etc/nsswitch.conf /etc/nsswitch.backup`
13. Type the command `cp /etc/nsswitch.nisplus /etc/nsswitch.conf`
14. Type the command `vi /etc/hosts`
15. Add the IP address and hostname for the NIS+ server. For example:


```
192.168.0.4    sun100
```
16. Type the command `/usr/lib/nis/nisclient -i -d aassdff.com -h sun100`
The following message will appear
Initializing client sun5 for domain "aassdff.com."

The following questions may be asked. If so, answer as shown:
Please enter the Secure-RPC password for root: type nisplus
Please enter the login password for root: type root123
17. Once initialization is done, you will need to reboot your machine.
On the NIS+ client, type the command

```
init 6
```

On the NIS+ Server:

18. Type the command `vi /etc/hosts`

19. Add the IP address and hostname for the NIS+ *client*. For example:
192.168.0.9 nisboy
20. Type the command **cd /etc**
21. Type the command **/usr/lib/nis/nispopulate -F**

On the NIS+ Client:

22. Type the command **ping nisboy**
*This pings the client. If successful, this produces the message **nisboy is alive**. This shows that the information came from the NIS+ server (it was specified in steps 18-21.)*

If this step fails, or if the next step fails, redo steps 18-21.

23. Type the command **ping -s nisboy**
This also pings the client, but if successful, it shows the client's IP address with the fictitious hostname "nisboy."
Press CTL + C to end this ping.

Working with NIS+ Tables

There are certain commands that are very useful when it comes to working with NIS+ tables. The NIS+ tables are stored in the **/var/nis/data** directory. These tables are binary files. Text editors and utilities like **vi**, **more**, and **cat** will not work on these files. Only NIS+ commands can be used on these files.

The niscat command

The **niscat** command works just like the **cat** command, except it only works on NIS+ tables. The **niscat** command supports the following options.

niscat <options> <NIS+Table>

<options>

- A** Displays all the data in a table.
- h** Displays header information before the contents of the table. Not all tables will display header information with this command.
- L** Tells the command to follow links. This only works if the table or object is linked to another object. For these examples that does not apply.
- M** Obtains this information from the Master Server only.
- o** Displays the internal representation of the objects.
- P** Follows the concatenation path to the requested object.
- s** Sets the separation character or symbol. For example, some tables might use a colon (**:**) as a separator for the columns in the table. The **-s** option can be used to specify this.
- v** Displays binary data on the command line.

The `nisbackup` Command

The `nisbackup` command is used to backup a NIS+ directory or the entire NIS+ domain. This can be used in conjunction with the `nisrestore` command.

The `nisrestore` Command

The `nisrestore` command is used to restore a NIS+ directory or the entire NIS+ domain. This can be used only if the `nisbackup` command was run previously. It is useful in case of a NIS+ server crash.

The `nisstat` Command

The `nisstat` command is used to show the status of a NIS+ server. This command shows what type of NIS+ server the host is. It also shows the directories that exist under the NIS+ server.

The lines after **Stat 'serves directories'**: show the default directories. If you have followed the lessons in this chapter, the defaults should be:

- `aassdff.com`.
- `org_dir.aassdff.com`.
- `groups_dir.aassdff.com`.

Among other information, this command also shows cache and storage statistics. It also shows the time the NIS+ server has been up since it began.

The `nisstat` command supports the following options:

```
nisstat <options> <NIS+_directory>
```

<options>

-H <hostname_of_NIS_server> This is the only supported option. It specifies the hostname of the NIS+ server that is being queried.

The `nisdefault` command

The `nisdefault` command shows the default settings for the NIS+ client. If the command is used without any options it will show the following pieces of information:

```
Principal Name : NIS+ hostname or username
Domain Name   : NIS+ Domain
Host Name     : Hostname of the NIS+ server
Group Name    : Group that the NIS+ client belongs to
Access Rights : The access rights that the principal has to the NIS+ server
Time to Live  : How long the NIS+ server's name will be considered valid
Search Path   : Domain to look in first for NIS+ information
```

Lesson 23.7 Working with NIS+ Tables

The NIS+ tables are in binary format. Because of this, the **niscat** command will be used to view the contents of the tables. The **niscat** command is similar to the **cat** command, except that it only works on NIS+ tables.

1. Log in as the root user on the NIS+ server.
2. Open a Terminal window.
3. Type the command **cd /var/nis/data**
4. Type the command **ls *org_dir**
The /var/nis/data directory contains all the tables in the NIS+ Root Master server for that server's domain (here, it should be the aassddff.com domain). These tables have the extension .org_dir on the end of their name.
5. Type the command **niscat -A passwd.org_dir**
The passwd.org_dir file was created from the original /etc/passwd file. These two files should match in most cases.
6. Type the command **niscat -A passwd.org_dir.aassddff.com**
This is the same niscat command, except that it uses the fully qualified domain name of the aassddff.com domain. This command would be used if a system had multiple domains and the passwd.org_dir table needed to be viewed from a specific domain.
7. Type the command **niscat -h networks.org_dir**
In this case the -h option shows the header information.
cname name addr and comment.
Understand that some tables will not show header information with the -h option.
8. Type the command **/usr/lib/nis/nisstat -H sun100**
The command nisstat -H displays statistics about the specified host. This should be the hostname of a NIS+ server.
9. Type the command **nisstat -H sun100 org_dir.aassddff.com**
This command shows the statistics for the org_dir directory. If a client has not been attached to the NIS+ server, don't be too surprised if this directory does not have any hits or statistics on its use.

Lesson 23.8 Removing NIS+

This lesson shows the reader how to remove NIS+ from a NIS+ client and server.

Note: If you did not install any NIS clients, start with the **On the NIS+ Server** section, below.

On the NIS+ Client

1. Log in as the root user.
2. Open a Terminal window
3. Type the command **/usr/lib/nis/nisclient -r**
4. Type the command **init 6**.
This reboots the workstation. If the workstation still seems to be interacting with the NIS+ server try these commands also:

```
/etc/init.d/rpc stop
rm /etc/.rootkey
/etc/init.d/rpc start
/usr/lib/nis/nisclient -r
```
5. If the client still seems to be interacting with the NIS+ server, copy the **/var/nis** directory to a tar file, then type the command **rm -rf /var/nis**
Understand that this is a very drastic step that could make it very difficult to re-perform the previous NIS or NIS+ lessons again.

On the NIS+ Server

6. Log in as the root user.
7. Open a Terminal window.
8. Type the command `/etc/init.d/rpc stop`
9. Type the command `rm /etc/.rootkey`
10. Type the command `/usr/lib/nis/nisclient -r`
11. Type the command `nisrmdir -f aassdfff.com`
12. Type the command `cp /etc/nsswitch.files /etc/nsswitch.conf`
13. Type the command `/etc/init.d/rpc.start`
14. Type the command `init 6`
15. If NIS+ still seems to be active on the server, copy the `/var/nis` directory to a tar file, then type the command
`rm -rf /var/nis`
Understand that this is a very drastic step that could make it very difficult to re-perform the previous NIS or NIS+ lessons again

Key Points to Remember

It is very important that the reader eventually try to work with two or three NIS servers and clients at one time. NIS and NIS+ are very difficult to master. This lesson only demonstrated how to work with a single NIS Master and NIS client. Most medium size companies use between 20 and 100 NIS and NIS+ servers. For system administrators who work in a small company, set up NIS instead of NIS+. NIS is a lot easier to understand and repair, compared to NIS+.

Chapter 24 Working with syslogd

Lessons in This Chapter

Lesson 24.1 Working with <code>/etc/syslog.conf</code>	24-3
Lesson 24.2 Logging TCP Connections	24-5

Introduction

The **syslogd** daemon is used to send error messages from the Kernel and software packages to a file in the `/var/adm` directory. The **syslogd** daemon can be configured to react to anything from simple warnings to critical error messages.

Syslog Error Messages

The **syslogd** daemon generates error messages from system utilities and the Kernel, and then puts them in a log file. This log file can be read to review error messages and warning messages, so that a server can be repaired or optimized. The **syslogd** daemon is configured by changing the `/etc/syslog.conf` file.

Error messages can be generated by four sources:

1. The Kernel
2. Processes generated by users
3. The **logger** command
4. System daemons (a daemon can be thought of as a program)

An error message can be sent to four possible locations:

1. To the system log. By default this is in the `/var/adm/messages` directory
2. To the console.
3. To other **syslogd** daemons on other servers on the network.
4. To users on a list.

The **syslogd** daemon can be started and stopped with the following commands:

```
/etc/init.d/syslog stop  
/etc/init.d/syslog start
```

The `/etc/syslog.conf` File in Detail

The `/etc/syslog.conf` file is a text file that can be modified to generate error messages, depending on the program and the severity of the error condition. For ordinary day-to-day operations, a system administrator might not want to have a lot of error and warning messages created. Except for troubleshooting and server optimization, a large number of error messages and warning messages can be useful.

Each line of the `/etc/syslog.conf` file has two parts, as shown in the following table.

What Generates The Error Message at What Warning Level	Where the Message Goes
facility.level	action field

As shown above, the *facility* entry specifies what generates the message, and the *.level* entry specifies at what warning level.

Here are some examples of lines in `/etc/syslog.conf`:

What Generates The Error Message at What Warning Level	Where the Message Goes
mail.crit	user12
user.err	dev/console

- The first line sends all mail-generated (**mail**) critical messages (**crit**) to the specified user (**user12**).
- The second line sends all user-generated (**user**) error messages (**.err**) to the console (**/dev/console**).

Let's examine the `/etc/syslog.conf` construction in more detail.

What Generates the Error Message (facility)	At What Warning Level (level)	Where the Message Goes (action)	Full Line in <code>/etc/syslog.conf</code>
mail	.crit	user12	mail.crit user12
user	.err	dev/console	user.err dev/console

As shown above, the **What Generates the Error Message at What Warning Level** parameter is broken down into two sub-parts < **facility.level** >.

The authorized entries for the **facility** (what can produce error messages) are:

- auth** Authorization programs (programs that authorize users on the system), such as **login** and **su**
- cron** The **cron** or **at** daemons
- daemon** System daemons, such as **telnetd** and **ftpd**
- lpr** Printing daemons, such as **lpd** and **lpadmin**
- mail** Received messages from the email programs, such as **mail** and **mailx**
- mark** Time stamped messages produced by the **syslogd** daemon
- news** Received error messages from news programs, such as USENET.
- syslog** Messages generated by the **syslogd** daemon itself
- user** Messages generated from programs that the user runs
- *** Anything (excludes messages from the **mark** facility)

The authorized entries for the **level** (how severe is error) are:

- debug** Very low level warnings used in debugging

info	Information messages, also used in debugging
notice	Information notices (not error messages but “good to know” messages)
warning	Warning messages
err	Error messages
crit	Critical error messages
alert	Very dangerous conditions (immediate attention required)
emerg	Emergency (the highest level of attention is needed now), usually generated when a server is about to crash

The **facility** shows what generated the error message, such as a bad Kernel process, the **mail** program, or corruption in a user's email.

The **level** determines at what threshold the error message should be generated. For example, should the **syslogd** daemon send an error message if a program has only a minor delay? Or should it send an error message only if a program completely dies? The **level** part answers these questions.

Entries for the **action field** (where to send the message) are determined as follows.

Once the **syslogd** daemon decides to send a message, it needs to have a destination for the message. There are four possible destinations:

- 1. /directory/directory/filename** Specify a directory and file name. This must be an absolute path name. For example **/home/user11/errorfile** is valid, but **myerrorfile** is not.
- 2. user,user,user** Specify a user or multiple users to receive error messages at their consoles. The users must be logged on at the time of the message. The users are separated by commas.
- 3. @host** Send the **syslogd** message to the **syslogd** daemon on another host. Requires a valid host name or IP address.
- 4. *** Everyone on the system will receive the message.

Lesson 24.1 Working with **/etc/syslog.conf**

In this lesson the reader will modify the **/etc/syslog.conf** file so that it gives debug level output to the screen. This will cause a lot of messages to be displayed on the screen, so make absolutely sure to return the **/etc/syslog.conf** file to its original settings after the lesson is done.

1. Log in as the root user.
2. Open a Terminal window.
To open a Terminal window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Tools menu item, then left click on the Terminal window icon.
3. Type the command **cp /etc/syslog.conf /etc/syslog.backup**
*This copies the **/etc/syslog.conf** file so it can be restored at the end of the lesson.*
4. Type the command **vi /etc/syslog.conf**
5. Add the following line to the **/etc/syslog.conf** file

for example,

```
mail.crit  ifdef ( ' DOG123 ', user12 , user14 )
```

In this example, let's say that the system sets the variable **DOG123** to **true** if a particular error condition exists (we won't worry here about which error that is). In this case:

- The M4 processor examines the variable **DOG123** and finds that it is true.
- Because **DOG123** is true, the *true condition* is executed. This means that all critical error messages (**.crit**) from the mail program (**mail**) will be sent to **user12**.
- If the variable **DOG123** were false, the M4 processor would execute the *false condition*. All critical error messages from the mail program would then be sent to **user14**.

Notice that this is the same format used by the `syslogd` daemon:

mail.crit is a <what generates the error message at what warning level > field

and

user12 is a < where the message goes > field.

For general system administration, it is not critical to know high level **ifdef** commands. Just understand that when the 'variable' is read (**DOG123**), the output of the <what generates the error message at what warning level > fields (**mail.crit**) is sent to the first location < where the message goes > (**user12**) if it is true, or to the second location < where the message goes > (**user14**) if it is false.

Logging TCP connections

Sometimes it is useful to log all connections to a server as a security precaution. If a hacker enters a server, this could help to find out where the hacker originated from.

To enable logging of all connections, put the following lines in the following files:

File	Line
<code>/etc/init.d/inetsvc</code>	<code>/etc/init.d/inetsvc -s -t &</code>
<code>/etc/syslog.conf</code>	<code>daemon.notice /mydirectory/tcpconnections</code>

Now all TCP connections will be sent to the **tcpconnections** file in the **/mydirectory** directory.

Lesson 24.2 Logging TCP Connections

In this lesson, TCP connections will be logged. The output will be sent to a file named **/mydirectory/tcpconnections**. Understand this will generate a large file over a short period of time and could slow down network connections on a slow system like a workstation.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command `mkdir /mydirectory`
4. Type the command `touch /mydirectory/tcpconnections`
*It does not matter how the **tcpconnections** file is created, just as long as the file is created. In this case the **touch** command creates a blank text file named **tcpconnections**.*
5. Type the command

```
echo "daemon.notice /mydirectory/tcpconnections" >> /etc/syslog.conf
```

*This command adds the line **daemon.notice /mydirectory/tcpconnections** to the **/etc/syslog.conf** file.*

6. Type the command **/etc/init.d/intesvc stop**
7. Type the command **/etc/init.d/inetsvc start**
8. Type the command **ping sun100** from any computer that can send ping messages to the Solaris workstation. If another system is not available, type the command **ping sun100** from this workstation. *Understand that **sun100** is this book's hostname for a system. If your system's hostname is different, type the command **ping <your_system's_hostname>**.*
9. Type the command **more /mydirectory/tcpconnections**
*You should see various log entries regarding the TCP connection that you made with the **ping** command.*

The logger command

The **logger** command is used to add a message to the system log file. The system log file is the **/var/adm/messages** file. Lines like "I took the server down today at 3:00 " can be very useful to enter into the system log file.

Some options for this command are:

```
logger <options> message
```

<options>

- f < file >** Use a message from the specified text file
- p <priority>** Set a priority (such as **crit** or **warn**)
- i** Record the process ID of the logger process
- t <tag>** Add the specified tag to the line **< mytag123 >**

To make the **logger** command work, add a line to the **/etc/syslog.conf** file. Here are two examples:

```
user.notice /mydirectory/mymessagefile or  
user.notice /var/adm/messages
```

Messages will be sent to the location specified for the user.notice field.

If a message needs to be added to the server's log files, it can be added with the **-p** option. For example if the **/etc/syslog.conf** file had the line

```
daemon.crit /mydirectory/kernelmessages
```

all daemons that had a critical error would send a message to a text file named **/mydirectory/kernelmessages**.

The command:

```
logger -p daemon.crit kernel was hung at 3:00 today, Mike Anderson
```

would add the message **kernel was hung at 3:00 today, Mike Anderson** to the text file `/mydirectory/kernelmessages`.

Basically, the **logger** command is used to send custom messages to the **syslogd** daemon. The **logger** command can be used by a script or on the command line by an administrator.

Key Points to Remember

This chapter discussed the **syslogd** daemon and the **logger** command in some detail, as well as how to log TCP connections. The **syslogd** daemon is primarily used to debug software. It is important that the system administrator know how the **syslogd** daemon works, to improve his or her troubleshooting techniques. The **logger** command is a nice utility to add entries to the **syslog** log files so that other system administrators will be provided with additional information by the previous system administrator.

Chapter 25 System Crash Analysis

Lessons in This Chapter

Lesson 25. 1 Using the <code>swap</code> Command	25-2
Lesson 25. 2 Using the <code>coreadm</code> Command	25-6
Lesson 25. 3 Using the <code>dumpadm</code> Command	25-9
Lesson 25. 4 Using the <code>adm</code> Command.....	25-10

Introduction

This chapter describes two concepts that are very important for a system administrator.

1. The first concept is dealing with swap space. Swap space is an area on the hard drive that the system uses as memory when actual memory is exhausted. The swap space uses a reserved area of the disk called a *swap file*. In an emergency, a system administrator needs to know how to create extra swap space in a hurry before the system crashes due to memory overflow errors. Managing swap space is an essential skill that should be mastered to avoid a system crash.
2. The second concept is how to troubleshoot what caused a server crash. This way, when the system is restored from tape (if necessary), the root of the problem can be tackled. The `coreadm`, `dumpadm` and `core` commands are the typical commands used for crash dump analysis.

The `swap` Command

When the Solaris operating system and software programs demand more memory than is physically available, Solaris 9 uses a part of the hard drive as *swap space*. Swap space refers to a disk partition or file that is used as if it were memory. The Solaris 9 operating system can use either a *swap file* or a *swap partition*. A swap file is a file that the `swap` command delegates to take swap requests. A swap partition (also known as the *swap file system*) is a slice on the hard drive that takes swap requests.

Which is better to use, a swap file or a swap slice? Some considerations are:

- A swap slice offers more efficient I/O. When a *swap file* is used, other read/write operations may be happening on the partition on which that file resides. This can slow down swap operations. When a dedicated *swap slice* is used, there are no other competing read/write requests to slow things down. However, it is harder to set up a swap slice, because you must reformat the disk to do so.
- A swap file offers less efficient I/O, as mentioned above. But it is much easier to set up a swap file. All you have to do is to create the file and then dedicate it to swap space.

Traditionally, slice 1 holds the swap partition. This file system is optimized for the task of reading and writing swap space to memory. If the swap partition starts to become full, the system administrator can add a *swap file* to create more swap space. A swap file, as mentioned before, is a file that has been designated for swap duty. A swap file is not as efficient as a swap partition, but it can add extra swap space to the system in a hurry.

Most experienced system administrators create a swap partition that is slightly larger than two times the size of the system's physical memory. This is important because when Solaris 9 crashes, it uses the swap partition to save a snapshot of the system's memory before a crash. If the swap partition is smaller than the physical memory on the system, only a partial dump of the system's memory will be performed.

Another piece of real world advice is to add more memory to a system when it starts using the swap partition too much. It is up to the system administrator to determine what is excessive use of the swap partition. Understand that when Solaris 9 uses the swap partition, the system's performance takes a dramatic hit. Also, try having several swap devices (swap slices and swap partitions). It's a good idea to use more than one physical device for swap space. This way, if one device fails (a hard drive dies) the swap system will not become disabled.

The **swap** command is used to manage swap space. It supports the following options:

swap <options>

<options>

- a** <swap_filename> Add swap space to the system, using the file <swap_filename>.
- a** </dev/dsk/c0t0d0s1> This uses slice 1 for the swap slice. You can specify other slices.

- d** <swap_filename> Take the named file out of the swap system. The file is not deleted.
- d** </dev/dsk/c0t0d0s1> Take the named slice out of the swap system. The slice is not deleted.

- l** List the current swap space on the system

- s** Summarize the swap space; this gives a different output than the **-l** option.

Lesson 25.1 Using the swap Command

In this lesson, readers use the **swap** command to view the current swap space and then add and delete swap space, using the swap file.

1. Log in as the root user.
2. Open a Terminal window.
To open a terminal window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Tools menu item, then left click on the Terminal window icon.
3. Type the command **swap -l**
That is swap (little L) This command shows the amount of swap space on the system.

Here is a sample output:

<i>swapfile</i>	<i>dev</i>	<i>swaplo</i>	<i>blocks</i>	<i>free</i>
<i>/dev/dsk/c0t0d0s1</i>	<i>136,1</i>	<i>16</i>	<i>1048928</i>	<i>1048928</i>

These columns show:

- *Swapfile: the fully qualified path name of the swap slice or swap file*
- *Dev: the device numbers (if it is a block special device) or all zeroes (if it not a block special device)*
- *Swaplo: the byte offset within the swap slice or swap file where the actual swap data begins*
- *Blocks: the total size of the swap slice or swap file, in 512-byte blocks*
- *Free: The number of 512-byte blocks in this area that are not currently allocated*

4. Type the command **swap -s**

Here is a sample output:

total: 109184k bytes allocated + 16368k reserved = 125552k used, 673288k available

This command shows the amount of space allocated, reserved, used, and available as swap space. Allocated space + Reserved space = Used space.

5. Type the command **mkfile 10m /myswapfile**
This command creates an empty file with 10 MB of space. It does not make the file be a swap file.
6. Type the command **swap -a /myswapfile**
This command uses the 10 MB file created earlier to add an extra 10 megabytes of extra swap space to the system with the **/myswapfile** file.
7. Type the command **swap -l**
That is swap (little L). Notice the difference in swap space?
8. Type the command **swap -s**
As above, this command shows the amount of space allocated, reserved, used, and available as swap space.
9. Type the command **swap -d /myswapfile**
This takes the file **/myswapfile** out of the swap space. The file itself is not deleted.
10. Type the command **swap -l**
That is swap (little L). Notice the difference in swap space.
11. Type the command **swap -s**
As above, this command shows the amount of space allocated, reserved, used, and available as swap space.
12. Edit the **/etc/vfstab** file and add the following line to it:
/myswapfile - - swap - no -
(type this line exactly as shown, with tabs between the entries).

This line in **/etc/vfstab** makes **/myswapfile** a permanent swap file for the system. When the **swap -a** command is typed on the command line, the specified file will be treated as swap space only until the next reboot. When the system reboots, the file will still be present, but it will not be seen as part of the swap system.

To make a swap file permanent, this entry must exist in the **/etc/vfstab** file. Without this entry, the next time the workstation reboots, the file **/myswapfile** will not be used for anything.

13. **OPTIONAL:** To add more swap space, another slice can be formatted with the UFS file system and added to swap space. To do this:

WARNING: the following substep uses the **newfs** command to create a new file system. All data currently residing on this file system will be lost. Back up all critical data before performing this step.

- a. Format the slice with the UFS file system. Here, we'll assume that this slice is **/dev/dsk/c0t2d0s0**. Type the following command:
newfs /dev/rdisk/c0t2d0s0
- b. Add the following line to the **/etc/vfstab** file:
/dev/dsk/c0t2d0s0 - - swap - no -

The disk slice **c0t2d0s0** is not the only disk slice that can be used. Any valid, un-mounted slice can be used for additional swap space. However, the slice needs to be formatted with the UFS file system.

14. Reboot the system.
15. Type the command **swap -l**
That is **swap** (little L).
16. Type the command **swap -s**
This command shows the kilobytes allocated, reserved used and available. The allocated and reserved swap space is the used swap space.

The `coreadm` Command

When a process or programs terminates in an abnormal way, it produces what is known as a *core file*. The core file is created in the working directory of the process. This core file can be analyzed to determine what went wrong with the program, and why it crashed. This is useful for software developers and software support specialists.

The `coreadm` command is configured with the `/etc/coreadm.conf` file. It can be configured in a very detailed way, so that specific programs produce a specific core dump. For more information about this, see the "Examining the `/etc/coreadm.conf` File in Detail" section, below.

Usually, a command is either restricted to the root user, or everyone can use the command in exactly the same way as the root user. The `coreadm` command is somewhat unique. A typical user can use this command a smaller set of options, while the root user has a larger set of options available.

The `coreadm` command supports the following options:

`coreadm <options>`

<code>-d</code>	Disables a specific type of core file, used with the <code>-e</code> option
<code>-e</code>	Specifies a type of core file
<code>global</code>	Uses core dumps with the global core pattern.
<code>process</code>	Uses core dumps on a per-processor pattern.
<code>global-setid</code>	Uses setid core dumps with a global core pattern.
<code>proc-setid</code>	Uses setid cord dumps with a per-process core pattern.
<code>log</code>	Generates a <code>syslog</code> error message when a user tries to generate a global core file.
<code>-g <pattern></code>	Supports the global arguments. For more information on the use of <code><pattern></code> , see the "Patterns Used with the <code>coreadm</code> Command" section, below.
<code>-i <pattern></code>	Uses the per-process core file name pattern.
<code>-p <pattern> <pid></code>	Sets the per-process core file name pattern to <code><pattern></code> for each process-ID that is specified. A superuser can apply the <code>-p</code> option to any process. Non-privileged users can only apply this option to processes that they own.
<code>-u</code>	Update the <code>/etc/coreadm.conf</code> file.
<code><pid></code>	The process ID number. Every process has a specific ID number associated with it.

Examining the `/etc/coreadm.conf` File in Detail

Figure 25.1 shows an `/etc/coreadm.conf` file in its default configuration.

```
#  
# coreadm.conf
```

```

#
# Parameters for system core file configuration.
# Do NOT edit this file by hand -- use coreadm(1) instead.
#
COREADM_GLOB_PATTERN=
COREADM_INIT_PATTERN=core
COREADM_GLOB_ENABLED=no
COREADM_PROC_ENABLED=yes
COREADM_GLOB_SETID_ENABLED=no
COREADM_PROC_SETID_ENABLED=no
COREADM_GLOB_LOG_ENABLED=no

```

Figure 25.1 The /etc/coreadm.conf File

Notice that the contents of /etc/coreadm/conf match the output of the `coreadm` command:

```

global core file pattern:
  init core file pattern: core
    global core dumps: disabled
  per-process core dumps: enabled
    global setid core dumps: disabled
  per-process setid core dumps: disabled
    global core dump logging: disabled

```

Each line has a specific purpose:

- global core file pattern:** Name of the global core file(s) if the global option is specified.
- init core file pattern: core** Name to be used if the `init` process generates a core file.
- global core dumps: disabled** Specifies if any global core files should be created. A global core file is saved when any process dies abnormally on the system.
- per-process core dumps: enabled** Used on servers with multiple processors, very useful for finding a bad processor.
- global setid core dumps:** Only used if setid core files are used. They are created with `rw-----` file permissions.
- per-process setid core dumps:** Only used if setid core files are used. They are created with `rw-----` file permissions.
- global core dump logging:** Enable or disable logging of global core dumps.

Patterns Used with the `coreadm` Command

The following patterns are used with the `coreadm -p <pattern>` command:

- %%** Literal pattern, follow whatever comes next.
- %f** Executable file name.
- %g** Effective GID (Group ID) of the process.

- %n** System node name (obtain with the command `uname -n`.)
- %m** Machine hardware name (obtain with the command `uname -m`.)
- %p** Process ID (obtain with the command `ps -ef | grep <program name>`.)
- %t** Decimal value of time.
- %u** Effective UID (UserID) of the process.

The adb Command

The **adb** command is used to examine the core files **unix.x** and **vmcore.x** that were saved by the **dumpadm** command. The **unix.x** and **vmcore.x** files should be in the `/var/crash/<hostname>` directory. The **adb** command can also work on **core** files generated when a program dies in an abnormal way.

The best advice when working on a server is to run the **adb** command on any **core** files that are created. Don't spend a great deal of time trying to analyze the stack traces and memory registry settings. A stack trace can be useful to have if you contact Sun. Otherwise, just figure out what program caused the **core** file and check that program's settings and installation.

The **adb** command supports the following options:

adb <options>

- k** Perform the analysis on the following files. Typical commands are:
`adb -k unix.0 vmcore.0`
`adb core`
- %r** Show the system's registry settings at the time of the crash.
- %c** Show the system's stack trace.
- %q** Quit the **adb** debugger.
- <\$msgbuf** Display error messages found in the crash files.

Lesson 25.2 Using the coreadm and adb Commands

In this lesson, you will use the **coreadm** command without any arguments, to examine the parameters set for the **coreadm** command. You will also create a core file for later crash analysis.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command `cat /etc/coreadm.conf`
*This command shows the contents of the **coreadm.conf** file. The **coreadm.conf** file is the configuration file that modifies the behavior of a core dump file.*
4. Type the command **coreadm**
*Without any arguments, the **coreadm** command displays the parameters set for **coreadm**.*
5. Type the command `mkdir /mytest`
6. Type the command `cd /mytest`
7. Type the command `admintool &`

The **Admintool** GUI should appear on the screen.

8. Type the command **pgrep admintool**
The **pgrep** command returns the PID (Process IDentification) number for the **Admintool** program. This should be a number like 203 or 604.
9. Type the command **kill -4 <PID_#_of_Admintool>**
The command **kill -4** sends a “dirty signal” to the **Admintool** that makes it crash and create a core file.
10. Type the command **admintool &**
The **Admintool** GUI should again appear on the screen.
11. Type the command **pgrep admintool**
The **pgrep** command will return the PID (Process IDentification) number for this new invocation of the **Admintool** program. This should be a number like 203 or 604.
12. Type the command **gcore <PID_#_of_Admintool>**
The **gcore** command creates a **core** file of a process. This file contains a snapshot of the program's memory setting when the program died. The only problem with the **core** file that you will see is that in this case, the program did not die in an ungraceful manner. Because of this, the **gcore** program does not create a true **core** file.
13. Type the command **adb core**
The command **adb core** starts the **adb** debugger on the **core** file.
You will see lines similar to:
core file = core -- program ``/usr/bin/admintool'' on platform SUNW,Sun-Blade-100
SIGILL: Illegal Instruction
(Ignore the last line.)

Notice the line that contains the name of the application (**/usr/bin/admintool**). This gives the name of the program that died. This can be a good starting point for figuring out what went wrong.

Note: When the **adb** debugger is running, it does not produce a prompt. You will be typing commands at the start of a blank line.

14. Type the command **\$c**
The **\$c** command shows the system's stack trace. How stack traces work is beyond the scope of this book. But if you need to contact an engineer at Sun about a problem, the stack trace can be very useful for analyzing a system crash.
15. Type the command **\$r**
The **\$r** variable shows the registry values at the time of the crash.
16. Type the command **\$q**
The **\$q** variable quits the **adb** debugger.

Quick Tip

The filenames shown above include ``uname -n`` in the absolute path. The nodename of the user's system will be substituted here. For the author's system, which has the nodename `sun100`, the files will be called:

```
/var/crash/sun100/unix.<x>  
/var/crash/sun100/vmcore.<x>
```

To understand this concept better, try typing the following commands as the root user:

- `whoami`
The output of this command will be `root` (the user's login name).
- `mkdir -p /mytestdir/`whoami``
- `cd /mytestdir/`whoami``
- `pwd`
The output of this will be `/testdir/root`

Notice that a directory was created based on the result of the ``whoami`` command, which appeared inside the two grave accent marks?

The path `/var/crash/`uname -n`` is doing the same thing. The result of the ``uname -n`` command becomes part of the pathname.

The `dumpadm` Command

The `dumpadm` command is similar to the `coreadm` command, except that it saves a snapshot of the contents of the system's memory into dump files (if possible.)

These files are:

```
/usr/kernel/drv/vmcore.<x>  
/usr/kernel/drv/unix.<x>
```

The `<x>` represents a version number for the generated dump, for example:

```
/usr/kernel/drv/vmcore.2  
/usr/kernel/drv/unix.1  
/usr/kernel/drv/unix.5
```

Any severe or fatal operating system error (such as faulty hardware, a bad device driver or a hung process) will cause the `dumpadm` command to save a memory dump to a hard drive. A message is also sent to the console, describing the error. By default, the `dumpadm` tries to use the swap partition. Another dump location can be specified if necessary.

After the contents of the system's memory have been saved, the system reboots. At this point in time, the contents are not in a readable format. After the system reboots, the **savecore** command automatically finds the crashed memory log and writes two files.

The files are:

- **/var/crash/`uname -n`/unix.<x>**
This file is a copy of the kernel core. During a crash, it is obtained from the **/dev/mem** directory.
- **/var/crash/`uname -n`/vmcore.<x>**
This file saves the name list information. During a crash, it is obtained from the **/dev/ksyms** directory.

The **dumpadm** command supports the following options:

dumpadm <options>

- c <kernel/all>** Configures the type of crash.
- d <dumpdevice/swap>** Sets the dump device. Use either **/dev/dsk/c#t#d#s#** or the keyword **swap**.
- m <min k/min m /min % >** Sets the minimum amount of free space that should be available after the **unix.<number>** and **vmcore.<number>** files are created. This is used to keep the **savecore** command from writing files that fill up the file system.
- n/-y** Specifies if the **savecore** command should be run on reboot (**-n no / -y yes**).
- r root-directory** Specifies an alternate root directory. For example:
 - r /mydirectory.** The dump files will be written to the directory **/mydirectory/var/crash/`uname -n`/**
- s save-directory** Specifies an alternate save directory. For example:
 - s /mydirectory.** The dump files will be written to the directory **/mydirectory.**

Lesson 25.3 Viewing the dumpadm Default Settings

This lesson only shows the default settings of the **dumpadm** command. Later in the chapter the system will be intentionally crashed. These settings determine how the information is collected from the crash.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command **cat /etc/dumpadm.conf**

*This command shows the default settings of the **dumpadm** command, in the following format:*

```
# dumpadm.conf
#
# Configuration parameters for system crash dump.
# Do NOT edit this file by hand -- use dumpadm(1m) instead.
#
```

```
DUMPADM_DEVICE=/dev/dsk/c0t0d0s1
DUMPADM_SAVDIR=/var/crash/sun100
DUMPADM_CONTENT=kernel
DUMPADM_ENABLE=yes
```

4. Type the command **dumpadm**

This command shows the same information, but in a more human-readable format:

```
Dump content: kernel pages
Dump device: /dev/dsk/c0t0d0s1 (swap)
Savecore directory: /var/crash/sun100
Savecore enabled: yes
```

Lesson 25.4 Using the adb Command

WARNING

In the following lesson, the server will be *intentionally crashed!*

- Do not do this on a production server!
- If you follow this lesson, make sure that you have backed up all critical file systems before you start the lesson.

In this lesson the reader will intentionally crash the operating system. When the system reboots it will create two crash files: **unix.0** and **vmcore.0**. The reader will then analyze the crash files with the **adb** command.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command **sync**
*The **sync** command makes sure that any information in memory is written to the disk. This prevents any disk cache behind information is left in memory and not recorded.*
4. Type the command **sync**
5. Type the command **sync**
*Running the **sync** command twice ensures that any cached read/write operations are written to disk.*
6. Type the command **halt -d**
*This command makes the operating system to do a panic dump (a series of messages, ending with the **OK** prompt).
The first time this command is run, the files **unix.0** and **vmcore.0** will be created. Each time **halt -d** is run, the file numbers are incremented by 1, so the next run would produce files called **unix.1** and **vmcore.1**.*
7. If the system does not reboot automatically, type the command **boot** at the OK prompt.
8. After the system reboots, type the command **cd /var/crash/<hostname>**
*The default crash directory should be **/var/crash/<hostname>**.
Type the command **dumpadm** to find the crash directory if **/var/crash/<hostname>** does not exist.*
9. Type the command **adb -k vmcore.0 unix.0**
*If the **vmcore.x** and **unix.x** files have a different number extension, type the command **adb -k <vmcore.x> <unix.x>**.*

This tells the adb debugger what files to work with.

10. Type the command **\$<msgbuf**

This shows all the messages that were in the memory buffer at the time of the system crash.

11. Type the command **\$c**

The \$c command shows the system's stack trace. How stack traces work is beyond the scope of this book. But if you need to contact an engineer at Sun about a problem, the stack trace can be very useful for analyzing a system crash.

12. Type the command **\$r**

The \$r variable shows the registry values at the time of the crash.

13. Type the command **\$q**

*The \$q variable quits the **adb** debugger.*

Key Points to Remember

Performing a system crash dump analysis is not a very easy thing to do. To really understand the **adb** command, you should get a book that covers this command in detail. Also, a very strong understanding of Sun SPARC hardware and memory is required to perform crash dump analysis. Most crash dump analysis is performed by a software engineer or software troubleshooting expert.

If you ever see that a **core** file has been created, this means that a process died abnormally at some point. To get information about this event, just type the command **adb core** in the directory. This should give the name of the program that died and created the **core** file. Most programs die from time to time, so it is not uncommon to find several **core** files in a server. The only area of concern is when the same program dies again and again.

Chapter 26 Volume Management

Lessons in This Chapter

Lesson 26. 1 A Quick Look at the Sun Volume Manager	26-6
Lesson 26. 2 Creating State Databases within the SMC.....	26-9
Lesson 26. 3 Using the <code>metadb</code> Command.....	26-13
Lesson 26. 4 Viewing and Deleting Metadevice State Databases from the SMC	26-16
Lesson 26. 5 Using the <code>format</code> Command to Prepare the 2nd Hard Drive	26-17
Lesson 26. 6 Creating a RAID 0 Volume	26-21
Lesson 26. 7 Mounting a Volume.....	26-26
Lesson 26. 8 Changing a Mount Option in the SMC.....	26-27
Lesson 26. 9 Adding extra Space to the Concatenated Volume	26-28
Lesson 26. 10 Deleting a RAID 0 Volume	26-30
Lesson 26. 11 Creating a RAID 1 Volume in the SMC.....	26-31

Introduction

One of the greatest problems with the standard UFS (UNIX File System) is that it only works over a single hard drive. What does a company do if it needs to use a 12 GB database file and a server only has two 10 GB hard drives installed?

Volume management is the concept of creating a *logical volume* over several disks. To answer the previous question, the company would create a logical volume of 20 GB, combining the two 10 GB disks. This is possible with the Solaris Volume Manager (SVM). The only thing that software and users *see* is a single large partition. The software and a typical user do not even know that this logical volume exists over two disks. In a nutshell, Solaris Volume Manager creates a partition over two or more disks that looks and acts just like any other partition to software packages and users.

SVM is run from within the Solaris Management Console (SMC). SVM also has command line tools that can be used just in case the CDE goes down and a GUI is not available. SMC supports several nice features, such as step-by-step wizards that guide a system administrator through all the steps in making various types of logical volumes.

SVM supports RAID 0, RAID 1, and RAID 5 volumes. It also supports hot swappable disks, soft partitions and transactional devices. These different types of volumes will be described in detail later in the chapter.

Sun Volume Manager

The Sun Volume Manager is used inside the Solaris Management Console. Figure 26.1 is a screen shot of the SVM inside the SMC. The Enhanced Storage icon (highlighted in the Navigation pane at the left of the screen) represents SVM.

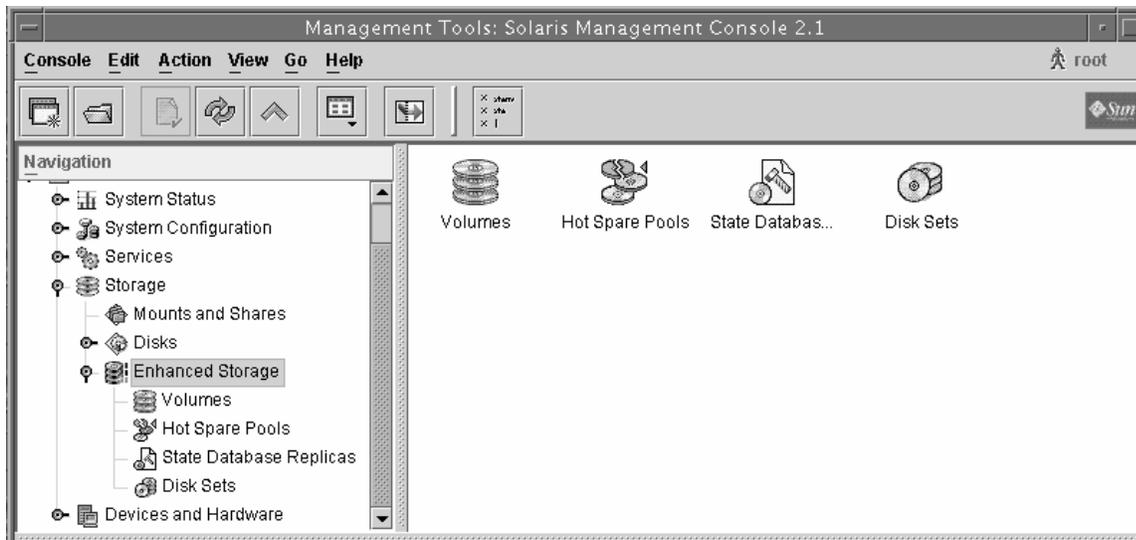


Figure 26.1 SVM Screen Shot

The SVM is also called the Enhanced Storage tool, as shown in Figure 26.1.

To run SVM:

1. Start the SMC.
*You will see the "This Computer" an icon for your system, with a name such as **sun100**.*
2. Left double click on the "This Computer" icon in the left pane.
 This Computer (sun100)
You can also left click on the Turner icon (the blue icon to the left of the "This Computer" icon).
3. Left double click on the "Storage" icon in the left pane.
 Storage
*A popup window might appear with the title "Log in: User Name."
This window asks for an authorized SMC user. Type in the username **root** and the root user's password (previous chapters have set up the root user's password as **root123**).*
4. Left double click on the "Enhanced Storage" icon in the left pane.
 Enhanced Storage
5. To exit from SVM and SMC, left click on Console in the Menu bar, then left click on the Exit menu choice.
*Another simple way to exit the SMC is to hold down the **CTL + Q** keys simultaneously.*

Volumes

The terms *Volume*, *Disk Volumes* and *RAID Volumes* are interchangeable. For readers who are familiar with Sun's DiskSuite Utility, the term *metadevice* is often loosely used to refer to a disk volume. Volumes created under SVM have a specific naming convention. All names must start with the letter **d** and must have a number between **0-127**. Valid volume names include **d10**, **d32**, and **d127**.

Most of the command line utilities can use the metadevice names as a form of shorthand for the names of physical devices. For example, you might see:

```
/dev/md/dsk/d23    The block device file for a volume
/dev/md/rdisk/d23 The raw device file for a volume
```

Remember that volumes are basically the virtual disks that are created by the SVM. They can be referenced by commands just as if they were real disk slices. For example:

- `mount /dev/md/dsk/d23 /mymountpoint` - mounts the **d23** volume to the directory `/mymountpoint`
- `fsck /dev/md/rdisk/d23` - checks the consistency of the **d23** volume
- `newfs /dev/md/rdisk/d23` - creates a new UFS file system on the **d23** volume

Disk Set

A *disk set* is a set of disks that are shared between two servers. Typically, the physical disks are external to the servers. Each server is connected to the disk controller on the disk array that contains the disks. Notice that this is not the same concept as a *shared disk* or a *networked disk*. When a disk is shared, there is only one server that controls that disk. Disks in a disk set are controlled by one or more servers.

Local Disk Set

A *local disk set* is a set of internal hard drives that can only be accessed by the server in which they are located. A local disk set contains the operating system for its server.

Figure 26.2 shows two servers. These servers share the disk set in the middle of the figure. Each server also has a local disk set, which only that server can access.

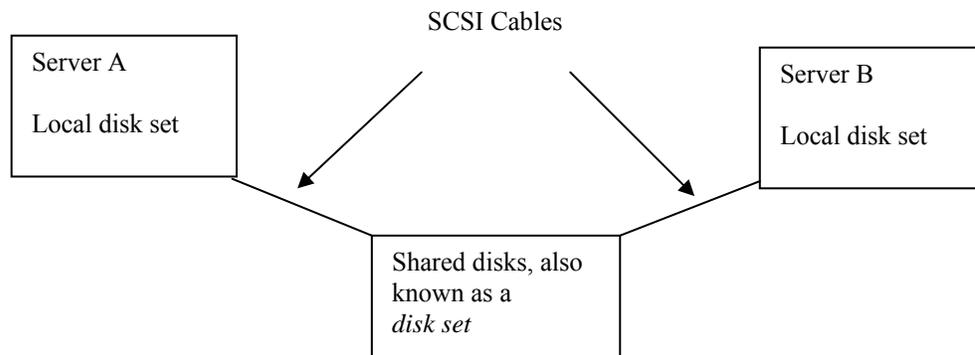


Figure 26.2 Illustration of a Disk Set

To access the shared disk, both of the servers in Figure 26.2 must have SVM installed. However, SVM is not designed to handle shared disk problems, such as having two servers simultaneously write to the same disk. Typically, software such as Veritas Cluster Server is used with both servers to handle this type of conflict on shared disks.

Hot Spare

A *hot spare* is a disk that just sits in an idle mode waiting for another hard drive to die. If another hard drive dies, SVM will automatically use the hot spare to take the place of the dead drive. A hot spare must be powered on and physically connected to a disk controller. Also, the hot spare must be of the same size or larger than the disk it is replacing. For example, a 10 GB disk can not be used as a hot spare for a 20 GB disk.

When trying to use a hot spare, SVM only looks at the size of the disk being replaced, not at the amount of information on that disk. Hot spares can only work on RAID 1 and RAID 5 devices. If a disk fails in a RAID 0 device, a new disk has to be installed, and the volume has to be restored from a tape or other type of backup.

Hot Spare Pool

A *hot spare pool* is a collection of hot spares. When a drive with RAID 1 (mirror) or RAID 5 (disk striping with parity) fails, a disk is selected out of the hot spare pool. This disk is then automatically rebuilt from the other disk's information.

Soft Partition

A *soft partition* can be thought of as a sub-volume. A soft partition divides a volume or partition into smaller pieces. A soft partition is the opposite of a volume. A volume collects several disks or slices into one logical disk. A soft partition lets you divide a physical disk into many smaller units. This can be useful if you want to have more than the usual number of slices on a disk. SVM supports up to 8192 soft partitions per disk set (the default is 128).

Transactional Volumes

The *transactional volumes* feature helps safeguard data integrity for a file system. When this feature is enabled, changes to a file system are logged to a transactional volume *before* they are written to disk. Let's say that something prevents a write operation from being completed (such as system crash). When the system reboots, any incomplete write operations will be discarded, using the information in the transactional volume. This ensures file system consistency, because only completed write operations are applied to the file system.

A transactional volume is composed of a *master device* and a *logging device*. The master device is the file system that is being read or written. The master device can be a disk slice, RAID 0 volume, RAID 1 volume or RAID 5 volume. The only drawback to using transactional volumes is that it slows down write operations to a certain degree.

RAID Levels

Most senior level administrators know the RAID (Redundant Array of Independent Disks) levels. But just as a review, here is a summary of some of those levels.

RAID 0 There are two different versions of RAID 0 volumes: RAID 0 Concatenated Volumes, and RAID 0 Striped Volumes. A single logical volume that is spread across two or more disks. For example, a RAID Level 0 disk could be created over disks `c0t3d0s0`, `c0t4d0s0` and `c0t5d0s0`, where each disk is 40 GB in size and the RAID 0 volume is 120 GB in size. RAID 0 reads and writes to the volume through all the disks at the same time. This RAID level offers the fastest read/write speed.

The greatest disadvantage to RAID 0 is that if a single disk in the RAID array fails, the entire volume becomes inaccessible. RAID 0 is also the hardest RAID to recover because if a RAID disk fails, it must first be replaced, and then all the information on it must be restored from backup media, such as a tape drive.

RAID 1 A type of volume management that mirrors one disk onto one more disks. With Solaris Volume Management, one disk can be mirrored onto several disks using RAID 1. Most system administrators use RAID 1 to mirror the system disk, which holds the operating system.

If a mirrored system disk fails, the server in theory should not crash. Unfortunately, sometimes the dying hard drive sends out a series of error messages that convince the operating system to halt itself, or to try an endless number of read/write operations, again and again. A main advantage of RAID 1 is

that both hard drives can service read requests, decreasing the amount of time to read data from the volume. The write time is not improved with RAID 1.

RAID 5 RAID 5 hard drives use what is known as a *striped partition*. Each hard drive in a RAID 5 volume contains parity information from each of the other drives in the volume. If one disk dies and is replaced, the remaining hard drives in the group supply the parity information necessary to restore the dead hard drive's information onto the new replacement drive. Read operations may be serviced by multiple disks in RAID 5, increasing the read time. Write operations are slower than any other operation because of the additional writing of parity information.

These RAID volumes work with many of the standard Solaris disk utilities, such as **fsck**, **df** and **du**. Unfortunately, the original definition of RAID is starting to become somewhat blurred, because most systems can perform one type of RAID partition on top of another type of RAID partition. For example, some companies use a RAID 0 volume over several disks. They then purchase another set of identical disks and use them to make a mirror copy (RAID 1) of the original disks. The result is a different type of RAID configuration (RAID 0+1).

There are also RAID systems that create RAID 0, RAID 1 and RAID 5 volumes at the *hardware* level. The operating system does not even know that the virtual hard drive is composed of several disks, because the hardware sends out virtual disk information that makes the virtual drive look like a giant physical drive. As technology increases with different volume management techniques, these different hybrids will stray further and further away from the original definition of RAID 0, RAID 1 and RAID 5.

One common misconception about RAID technology is that the hard drives are redundantly protected and that there is no need for backup media such as tape drives. This is a very dangerous idea! If a piece of software writes bad data onto a file, or if files become corrupted due to software or operating system errors, no RAID technology in the world can restore the original good files. RAID technology will faithfully save any file changes made. So, if a 10 GB database file is suddenly saved as a 0 byte file, RAID will not know there is a problem and will save the 0 byte file without hesitation. Also, if a file is deleted, it can not be recovered from a RAID volume. RAID's strength is that it can guard against hardware failure and therefore maintain server up-time. A tape drive (or other backup medium) is still needed for the classic role of saving and restoring files.

State Databases

State databases are used by the Solaris Volume Management tool to store information about the system's volume devices (such as disk sets, volumes and hot spares). A state database keeps track of all changes made to the volume devices. SVM uses multiple identical databases, for redundancy. If one state database goes bad, SVM will work with the remaining good state databases. When the system starts or reboots, SVM reads these databases to get full information about the volume structure currently in place. Once Solaris 9 has completed its boot process, SVM writes any new changes to the state databases.

Quick Tip

The following terms are interchangeable:

Solaris Volume Management state database replica
Solaris Volume Management state database
replica (within the context of “state database replica”)
metadevice
database replica

Technically, a state database replica is a copy of an existing state database. After the replication process is done, the replica and the original state database are identical. For ease of reading, the remainder of this chapter will use the terms *state database*, *state database replica* and *database* instead of the rather awkward term *Solaris Volume Management state database replica*.

A system needs a minimum of three state databases. The reason for this is that if the system crashes while one of the databases is being written, only one state database will become corrupt. When the system reboots, SVM compares the state databases and uses the ones that are consistent to work with the virtual file system. The theory is that the bad database will be the *odd man out*. A warning message about the bad database is sent to the system administrator.

State databases can be saved on a disk slice, on a partition inside a volume or on a partition inside a transactional volume. The only exception is that a replica can not be set up on the **/usr** slice, the **swap** slice (slice 1) or the root (**/**) slice (slice 0.)

State databases can also be saved over several disks. For example, if a server has three hard drives, all three hard drives can hold state databases, so that if one hard drive dies, the server’s virtual file system will not be damaged.

Each state database has a size of 4 MB, or 8192 sectors of 512 bytes each (default size). If a slice is reserved for state databases, the slice must have at least 4 MB of space for each state database. It is a good idea to allocate more than 4 MB per database, to handle any future growth of the database in the future. Just as a quick side comment, under Solaris 9 a slice can not be smaller than 10 MB in size.

SVM has an upper limit of 50 state databases per disk set. Databases can be added or subtracted at any time.

Lesson 26.1 A Quick Look at the Sun Volume Manager

SVM runs inside of the Solaris Management Console. The first time the SMC starts it may take several minutes to analyze the system and set up its own configuration files. This lesson demonstrates how to start the SMC and where the Solaris Volume Management tool is inside the SMC.

Only the root user can use SVM to change server settings. An ordinary user can view items in the SMC, but can not change the server’s settings.

1. Log in as the root user.
2. Right click anywhere in unoccupied desktop space.
The Workspace menu appears.

3. Left click on the Tools menu item.
4. Left click on the "Solaris Management Console" icon.
To start the SMC from the command line, type the command `/usr/sbin/smc &`
5. Left double click on the "This Computer" icon in the left pane.

 This Computer (sun100)

You can also left click on the Turner icon (the blue icon to the left of the "This Computer" icon).

6. Left double click on the "Storage" icon in the left pane.

 Storage

A popup window might appear with the title "Log in: User Name."

*This window asks for an authorized SMC user. Type in the username **root** and the root user's password (previous chapters have set up the root user's password as **root123**).*

7. Left double click on the "Enhanced Storage" icon in the left pane.

 Enhanced Storage

The following four icons should appear in the right pane:



Volumes

Volumes

This shows the volumes created with SVM.



Hot Spare Pools

Hot Spare Pools

This shows the hard drive or drives waiting to take the place of a dying hard drive.



State Database

State Database

This shows the small databases that hold volume configuration information.



Disk Sets

Disk Sets

This shows any hard drives shared between two computers

8. Explore the SVM, but do not make any permanent changes to the system. Do not be too surprised if clicking the SVM icons shows little or no information. At this point, nothing has been created with SVM.
9. To exit from SVM and SMC, left click on Console in the Menu bar, then left click on the Exit menu choice.
Another simple way to exit the SMC is to hold down the CTL + Q keys simultaneously.

Icons Associated with Volume Management

This section describes some of the icons shown in through Lesson 26.1.



Volumes

Volumes: This tool is used to view, create, delete and change RAID 0, RAID1, and RAID 5 volumes. It also handles transaction (logging) volumes and soft partitions. The Volumes tool also allows the user to create state databases, which are used to store information about the volumes.



Hot Spare Pools

Hot Spare Pools: This tool is used to view, create, delete and work with hot spare pools. A hot spare pool is used when a disk in a RAID volume goes bad. When this happens, a hot spare drive from the hot spare pool is used to replace the bad disk.



State Databases...

State Database Replicas: This tool is used to show the state database replicas. The databases are shown with the following codes and terms:

	Name	Name of the slice that holds the state database replica
Disk Set	The disks that are being managed by the state database replica	
State	Condition of the state database replica	
	a	Replica is actively being used
	c	Replica is referenced in the <code>/etc/lvm/mddb.cf</code> file
	D	Replica has problems with data blocks
	F	Replica has format problems
	l	Locator for the replica was read successfully
	m	Replica is the master
	M	Replica has problem with master blocks
	p	Replica's location was patched in the Kernel
	R	Replica suffers from device read errors
	S	Replica is too small for the current database
	W	Replica has a device with write errors
	u	Replica is updated correctly
Start Block	Block where the replica first starts.	
Size	Default size for a replica is 4 MB	



Disk Sets

Disk Sets This shows the current disk sets. This tool shows hard drives that are being shared between two computers.

Creating State Databases

Remember that the terms *state database replica* and *state database* describe the exact same thing. There is no logical or physical difference between them. The term *replica* is used for state databases that were created after the original state database was created.

Just as a quick review, each state database occupies 4 MB of space, or 8192 disk sectors (each disk sector is 512 bytes in size). There must be at least 3 state databases on a file system, with a maximum of 50. A state database can be saved on a volume, disk partition or logging device. The only slices that can not be used are slices that already have a file system with files and directories. This means that slices that are mounted under the root (/), **swap** or **/usr** slice can not be used. Also, if a slice does not have a file system created, a state database will not populate that slice. A state database cannot be added to a mounted slice. So, create the file system first, but don't mount the file system until you have created the state databases.

There are two ways to create, manage and delete state databases. As mentioned above, you can use the Enhanced Storage tool in SMC. To view this tool, review Lesson 26.1 above. The Enhanced Storage tool can create a log file of all the commands that have been used to create, modify and delete state databases. This tool provides several installation wizards that can simplify the installation of state databases on the system's file systems.

The other way to create state databases is to use command line utilities such as **metadb**. The **metadb** command can be used to create, display, and delete the state databases. Be careful: this tool has some advanced features that a junior level system administrator should avoid. Using **metadb**, it is possible to destroy all the state databases that work with a disk set, and to destroy the disk set in the process.

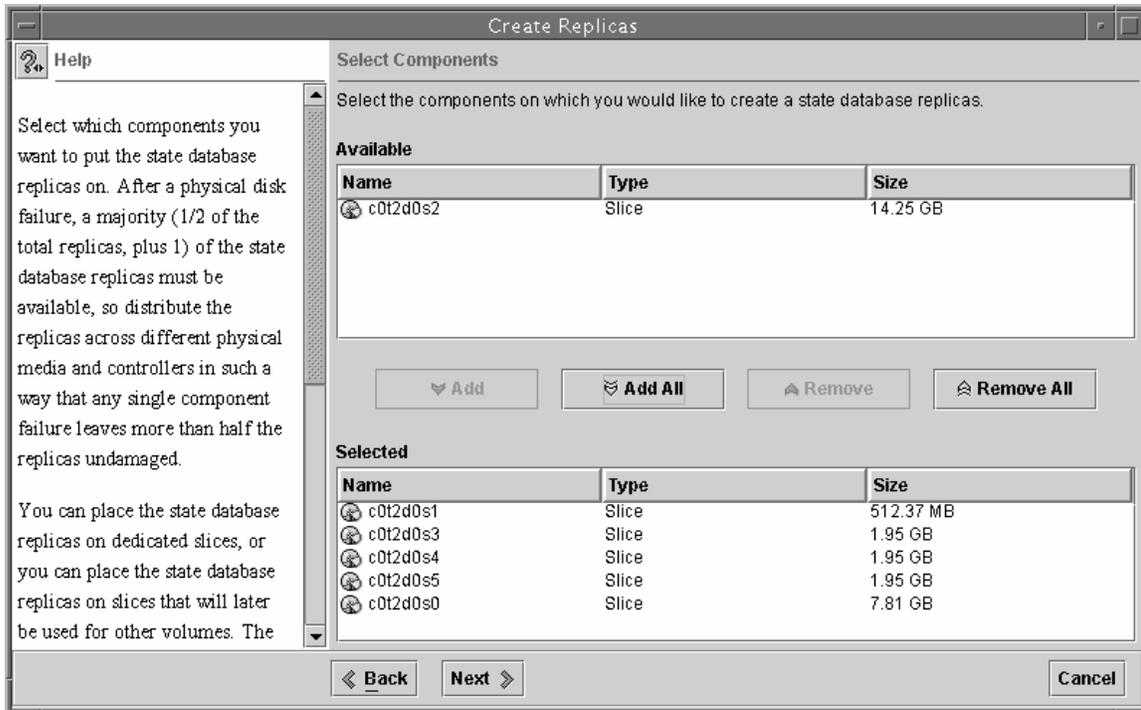
Lesson 26.2 Creating State Databases within the SMC

In this lesson, three state databases will be created on the system disk. This lesson may have to be modified to match the system setup on the reader's test workstation. The key point is to make three state databases on a slice or hard drive. Remember that having more state databases decreases the chances of having a problem with a corrupted database. The upper limit for state databases is 50 per volume or disk set, so don't get carried away.

1. Log in as the root user.
2. Open the SMC.
To open the SMC, right click anywhere in unoccupied desktop space. In the Workspace menu, left click on Tools, then left click on the "Solaris Management Console" icon.
3. Left double click on the "This Computer" icon in the left pane.
 This Computer (sun100)
You can also left click on the Turner icon (the blue icon to the left of the "This Computer" icon).
4. Left double click on the "Storage" icon in the left pane.
 Storage
*A popup window might appear with the title "Log in: User Name."
This window asks for an authorized SMC user. Type in the username **root** and the root user's password (previous chapters have set up the root user's password as **root123**).*
5. Left double click on the "Enhanced Storage" icon in the left pane.
 Enhanced Storage
6. Left click on the State Database Replicas icon. 
7. Left click on Action in the Menu bar.
8. Left click on the Create Replicas... menu choice.
A window entitled "Create Replicas" will appear



9. Left click on the Next > button.
*This window lets you select disk sets, but none have been created so far.
The next screen shows the disk slices that can be used for state databases.*

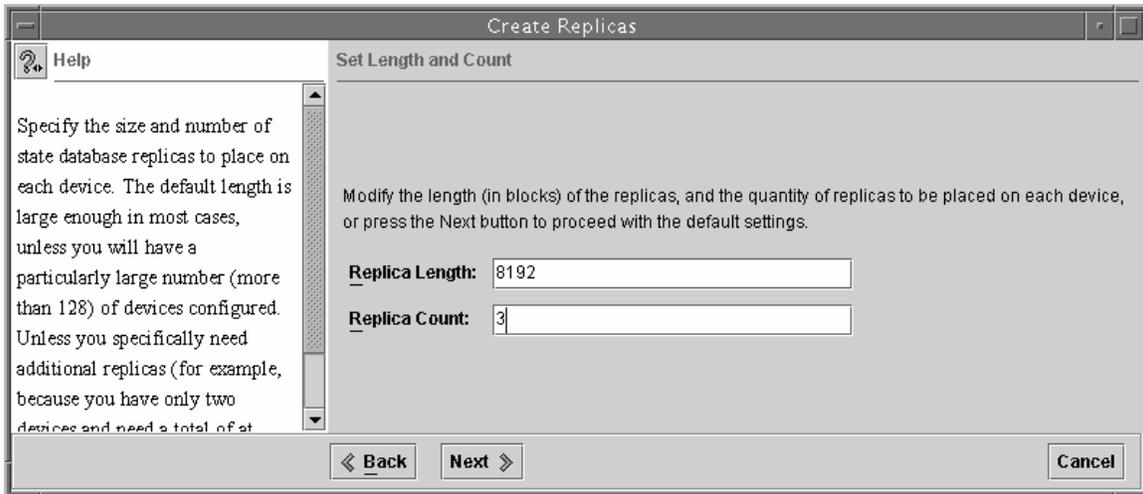


10. Select any slice in the Available: window at the top. Then, left click on the ▼ Add button to move it to the Selected window at the bottom. Continue doing this with *each available slice*, until all of them have been moved to the Selected window.

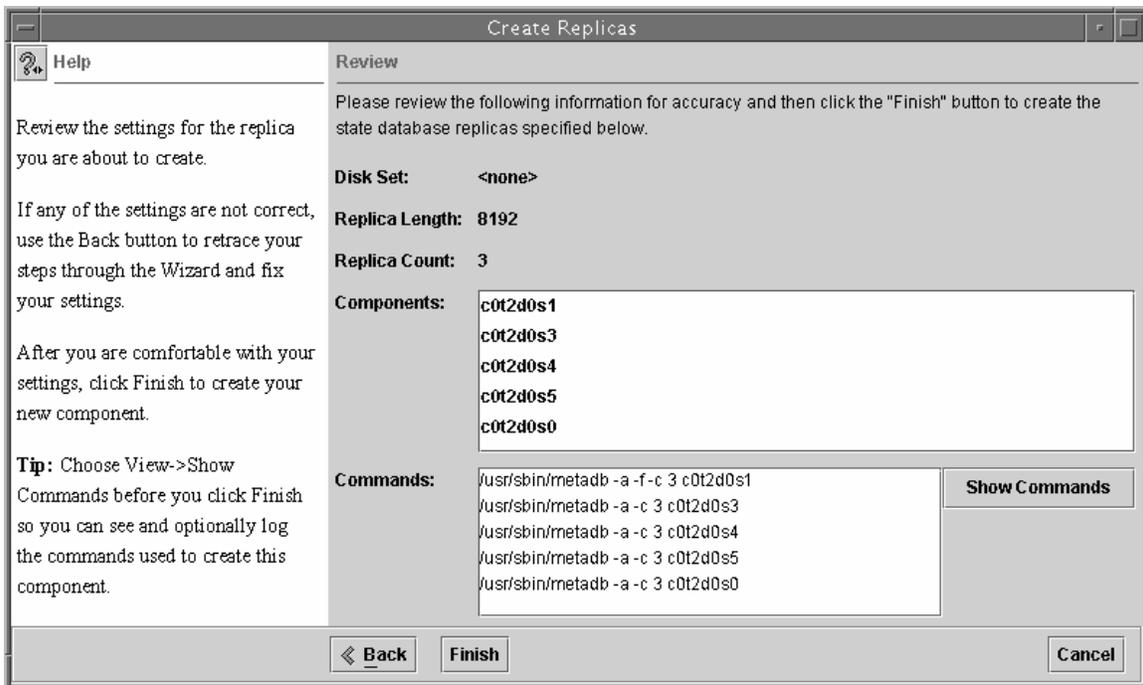
Each state database takes up 4 MB of space. Even if you create 3 state databases each on 10 separate slices, that would only be 120 MB of disk space. That is not too much space, considering that most modern hard drives are 20 to 40 GB in size, and this lesson is being performed on a test workstation.

If for some reason your test workstation has limited disk space, select fewer slices. If an error message appears about any slice, do not worry This might happen because you have selected a prohibited slice You can ignore the error message. Select and add as many slices as possible, using the ▼ Add button.

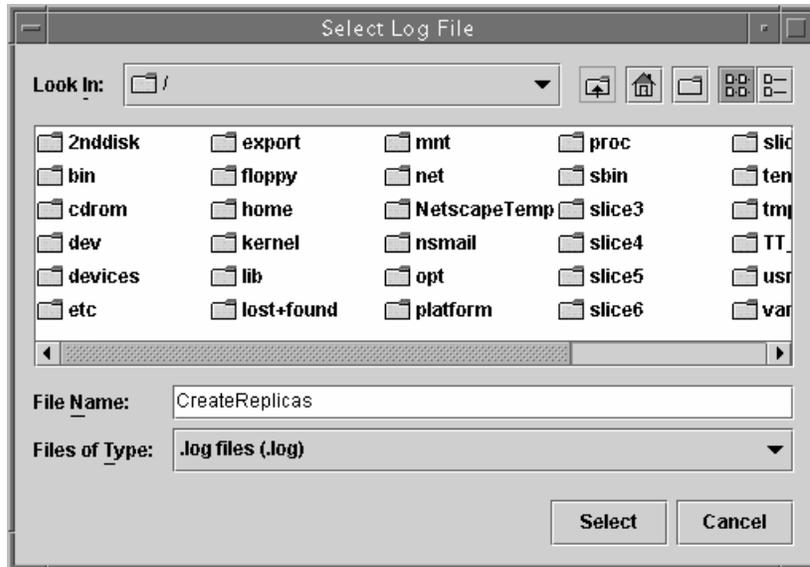
11. Left click on the Next ► button
The next screen asks for the number of state databases and the size of the state database to put on the slices selected in the previous screen.



12. Keep the default size of 8192 blocks in the Replica Length: text box. Change the number of state databases to 3 in the Replica Count:text box.
13. Left click on the “Next >” button
The next screen is a summary screen that shows the components of the state database that will be produced. It also shows the command line commands that will be used to create it.



14. Left click on the Show Commands button (at the lower right of the window).
A smaller popup window should appear that lets the administrator log the commands used to a text file.



15. Use the Look In box to navigate to the root (/) directory.
16. In the File Name: text box, type **CreateReplicas**
17. Left click on the Select button.

*A popup window might appear with the warning message **A Tool loaded from su100:898 wishes to write to the file located at /createreplicas**. This is just a security warning. If this happens, left click on the checkbox next to the **Don't** ask me again about permissions from this location sentence, and then left click on the Grant button.*

18. Open a Terminal window.
19. Type the command **cat /CreateReplicas**
20. To find out more about the **metadb** command, type the command **man metadb**

*What does the **metadb** command do? As can be seen by this exercise, the Enhanced Storage tool is a GUI that runs the **metadb** command in the background to create the state databases. The **metadb** command is covered in detail after this lesson.*

In the SVM window, the right pane shows all the state databases. Notice that:

- *The databases are shown with the state **a u**. This indicates that the state databases are active and up to date.*
- *These databases are not currently being used by a disk set, so they have the term **<none>** for the name of the disk set used.*
- *The size of each state database should be 3.0 MB.*
- *The Information pane shows the code letters used for a state database.*

21. Left click on Console in the Menu bar.
22. Left click on Exit.

Using Command Line Tools with State Databases

It is absolutely critical that a system administrator be able to create, manage and update state databases from the command line. The SMC Enhanced Storage tool is a very convenient way of creating state databases, but what happens if the graphical interface is not available? Understand that unless 50% of the state databases on a server are usable, the server will not boot! Instead, the system will stop in single user run level (also known as

maintenance mode). In this run level, a graphical user interface is not present, and the server must be brought back up using command line commands.

Understanding the `metadb` command

The `metadb` command is the primary command used to manage state databases from the command line. This command can create, modify and delete the state databases.

The `metadb` command supports the following options:

`metadb <options> <slice>`

`<options>`

- `-a` Add a new metadvice database replica.
- `-c` Used only with the `-a` option. It is used when more than one metadvice state database or replica is going to be placed on a slice.
- `-f` Force the desired `metadb` command to work. There are times that `metadb` will hesitate to perform a command. The `-f` option forces `metadb` to perform an action. This overrides some of the safeguards built into the `metadb` command.
- `-h` Displays a brief text help message.
- `-i` Displays the meaning of each of the letter keywords used to describe the metadvice state database.
- `-k` Avoid this option. It changes the default Kernel file used with replica information. This is a bad idea, so just leave this option alone!
- `-l` Used to set the size of the database. The default size is 8192 disk sectors, (4 MB). Unless there is some type of error message stating that the metadvice database replica is too small, don't use this option. The default size of 4 MB is adequate.
- `-p` Another option to avoid. It updates the `/kernel/drv/md.conf` file with entries from the `/etc/lvm/mddb.cf` file. It is better to create a metadvice state database by hand than to try to automatically update the `/kernel/drv/md.conf` file.
- `-s` The disk set that these metadvice state databases will be used with.

`<slice>`

`/dev/dsk/c#t#d#s#` This can be the logical device name for the disk slice or
`c#t#d#s#` the name of the slice, shortened to `c#t#d#s#`.

Lesson 26.3 Using the `metadb` Command

As mentioned above, the `metadb` command is used to create, manage and delete state databases. In this lesson, readers will examine the state databases that were created with the SMC Enhanced Storage tool in the previous lesson. The metadvice state databases will be deleted and then created using only command line tools only.

Note: This lesson can only be performed on metadevice state databases that are not attached to a disk set. If metadevices are using these databases, do not try this lesson. Also note that metadevices can only be created on slices that are not currently mounted.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command **metadb**

The command **metadb** without any options shows the current metadevice state databases.
The output from the author's SunBlade 100 is as follows (yours may be different):

flags	first blk	block count	
a u	16	8192	/dev/dsk/c0t2d0s0
a u	8208	8192	/dev/dsk/c0t2d0s0
a u	16400	8192	/dev/dsk/c0t2d0s0
a u	16	8192	/dev/dsk/c0t2d0s1
a u	8208	8192	/dev/dsk/c0t2d0s1
a u	16400	8192	/dev/dsk/c0t2d0s1
a u	16	8192	/dev/dsk/c0t2d0s7
a u	8208	8192	/dev/dsk/c0t2d0s7
a u	16400	8192	/dev/dsk/c0t2d0s7

4. **IMPORTANT! PRINT or COPY** the list of slices for your system that is shown in the previous step.
This list will be used later in the example. Failure to save this information will cause problems!!!

5. Type the command **clear**
This clears all text from the terminal window.

6. Type the command **metadb -i**
The **-i** option displays the meaning of the letter keywords used to describe the metadevice state database.

7. Type the command **clear**

8. Type the command **metadb -h**
The **-h** option displays a brief text help message. This is quicker than using the **man metadb** command to obtain basic help information.

9. Type the command **metadb -d /dev/dsk/<first_slice_containing_a_md_database>**.

Only remove the state database replicas from the **first** used slice.

For example, the author's list of slices containing metadevice state databases is:

```
c0t2d0s0
c0t2d0s1
c0t2d0s7
```

The first used slice here is **c0t2d0s0**, so the command the author typed here is:

```
metadb -d /dev/dsk/c0t2d0s0
```

10. Type the command **metadb**
After the deletion, the first reported slice will be gone.
On the author's system, the output is:

flags	first blk	block count	
a u	16	8192	/dev/dsk/c0t2d0s1
a u	8208	8192	/dev/dsk/c0t2d0s1
a u	16400	8192	/dev/dsk/c0t2d0s1
a u	16	8192	/dev/dsk/c0t2d0s7
a u	8208	8192	/dev/dsk/c0t2d0s7
a u	16400	8192	/dev/dsk/c0t2d0s7

11. Repeat the **metadb -d** command in step 9 (and repeat step 10, if desired), for each slice that contains metadevice state databases. Continue until all the metadevice state databases have been deleted from the system. If this command would delete the last couple of metadevices, an error message will appear. If this happens, use the **-f** option to force the **metadb** command to perform this action.

12. Type the command **metadb**
*The command **metadb** without any options shows the current metadevice state databases. There should be none present!*
13. Find the list of slices that you made in step 4 (if you don't have that list, see the next step for suggestions).
14. Based on the list of slices from step 4, use the command
metadb -a -c 3 <slices_from_your_list>
to recreate all of the deleted slices, one by one.

On the author's system, these three commands would be used:

- **metadb -a -c 3 c0t2d0s0**
- **metadb -a -c 3 c0t2d0s1**
- **metadb -a -c 3 c0t2d0s7**

If error messages appear, see step 16.

If you didn't make a list of slices in step 4, try one of the following options:

- *If possible, scroll back up in your Terminal session until you find the output of the **metadb** command in step 3. You can then use this output as the list of slices to put back.*
 - *If this is not possible, redo the previous lesson and create the metadevice state databases from the SMC Enhanced Storage tool. Then, start this lesson again. Be sure to copy or print the information in step 3!*
15. To verify that the state database replicas have been recreated, type **metadb**
The output should be the same as in step 3. If so, skip the next step.
 16. If the databases were not successfully restored in step 14, try the **metadb** command with the **-f** option. Then, repeat step 15 to verify that the databases were created.
*This forces the **metadb** command to create a database on a slice. For the first slice on the author's system, this would be:*
metadb -a -f -c /dev/dsk/c0t2d0s0
 17. Type the command **cat /etc/lvm/mddb.cf**
*The **/etc/lvm/mddb.cf** file is used by the system to record the location of the metadevice state databases.*

This file should not be edited by hand. The contents of the file can be different, depending on the system's architecture. The display below is from a SunBlade 100 with two IDE disks and an IDE CDROM drive.

```
#metadevice database location file do not hand edit
#driver minor_t daddr_t device id checksum
dad      8      16      id1,dad@AWDC_AC35100L=WD-WT3900326085/a -2869
dad      8      8208     id1,dad@AWDC_AC35100L=WD-WT3900326085/a -11061
dad      8      16400    id1,dad@AWDC_AC35100L=WD-WT3900326085/a -19253
dad      9      16      id1,dad@AWDC_AC35100L=WD-WT3900326085/b -2871
dad      9      8208     id1,dad@AWDC_AC35100L=WD-WT3900326085/b -11063
dad      9      16400    id1,dad@AWDC_AC35100L=WD-WT3900326085/b -19255
1        2        3        4        5
```

Reading left to right, the fields are as follows (numbers have been added above):

1. *Symbolic driver name of the disk.*
2. *Minor unit number associated with the disk.*
3. *Starting block of the metadevice state database.*
4. *Physical device name of the disk. The end of this field shows the disk block address:
/a = slice 0, /b = slice 1, /c = slice 2 (never used), /d = slice 3, e = slice4, /f=slice5, /g=slice6, /h=slice 7*
5. *Checksum*

Remember, do not edit this file by hand. It can be updated with the **metadb** command. In this example the **mddb.cf** file only shows the **c0t2d0s0** and **c0t2d0s1** slices. Remember, do not edit this file by hand.

Monitoring and Deleting Metadevice State Databases from the SMC

The **metadb** command lets you display and delete the state databases from the command line. It is also possible to do these things in a GUI environment with the Enhanced Storage tool.

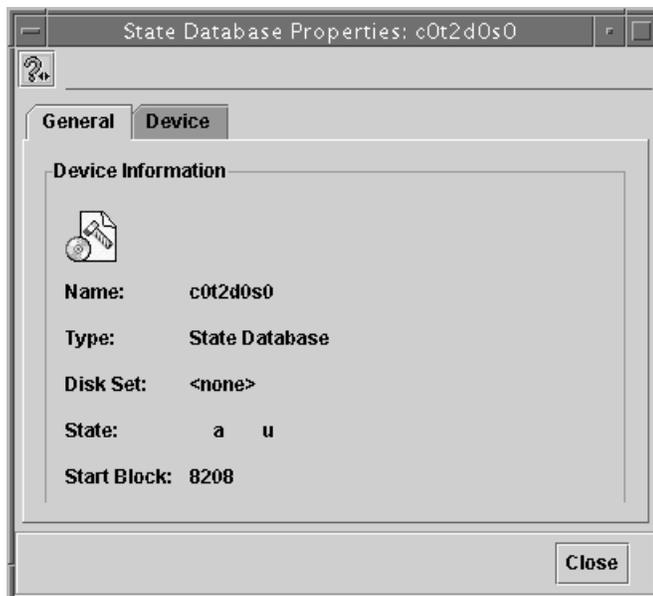
Lesson 26.4 Viewing and Deleting Metadevice State Databases from the SMC

In this lesson the student will view and delete the state databases that were created from previous lessons, using the SMC.

1. Log in as the root user.
2. Open the SMC
To open the SMC, right click anywhere in unoccupied desktop space. In the Workspace menu, left click the Tools menu item, then left click on the "Solaris Management Console" icon.
3. Left double click on the "This Computer" icon in the left pane.
 This Computer (sun100)
You can also left click on the Turner icon (the blue icon to the left of the "This Computer" icon).
4. Left double click on the "Storage" icon in the left pane.

*A popup window might appear with the title "Log in: User Name."
This window asks for an authorized SMC user. Type in the username **root** and the root user's password (previous chapters have set up the root user's password as **root123**).*
5. Left double click on the "Enhanced Storage" icon in the left pane.

6. Left click on the "State Database Replicas" icon. 
7. Quickly left double click on one of the metadevice state databases shown in the right View pane.
This will bring up the State Database Properties window.



At the top of the window, the *General* tab shows the same information as the *View* pane. The *Device* tab shows information about the device that the metadevice state database resides on.

8. Left click on the Close button.
9. Select all of the state databases shown in the right View pane.

This should bring up a small menu with the following items:

- | | |
|------------------------|---|
| Show Commands | - Used to log commands to a log file |
| Create Replicas | - Opens the C reate Replicas... wizard |
| Delete | - This command will delete all the replicas on the slice, not just the highlighted replica. |
| Properties | - Displays the replica's "State Database properties" window again. |

After all the replicas are highlighted, click on *Delete*. Because the state databases are not associated with a disk set, they can be safely deleted and created without creating a problem.



10. Left click anywhere in unoccupied space to close this menu.
11. Left click on **A**ction.
12. Left click on **C**reate Replicas.
13. Create 3 replicas on every slice possible, with a size of 8192 blocks. Follow the same steps that were presented in lesson 26.2, but this time do not save the commands in a log file.
Lesson 26.2 demonstrated how to create state databases with the SMC. Review Lesson 26.2 if you run into any kinds of problems with this step.
14. Left click on **C**onsole.
15. Left click on **E**xit.

Working with RAID 0 Volumes

The Solaris Volume Management software creates what are known as *volumes*. A RAID 0 volume can have one of two possible forms:

- **Concatenated Volume.** This stores its data in one large volume that is composed of several smaller partitions. When one partition fills up, the next partition is used.
- **Striped Volume.** This uses all of the disk drives in concurrent read/write operations.

One of the principal advantages of a Concatenated Volume is that **might be** possible to still read files from a concatenated volume if one of the disks that makes up the volume dies (if this disk did not have data on it). However, this is not a sure thing.

The advantage of a striped volume is that it can perform read and write operations much faster than a concatenated volume. The disadvantage of a striped volume is that if one disk dies, the entire volume dies and the data can never be recovered from the volume. A tape restore would typically be needed to recover the data.

Lesson 26.5 Preparing the Hard Drives for Future Lessons

This lesson prepares the hard drives on your system for the lessons that follow. It can be followed on a system that has two hard drives, or on a system that has only one hard drive. Be sure to read the warning below before starting the lesson.

WARNING

- This lesson will destroy any data on the *second hard drive*. Make sure you back up any critical files from the second hard drive before performing this lesson or subsequent lessons.
- If your system has only one hard drive, *and* you installed Solaris 9 using the instructions in Chapter 2, the system will *not* be damaged while following this chapter's lessons.
- If your system has only *one hard drive*, and you did *not* install Solaris 9 according to the instructions in Chapter 2:

Make sure that no critical file systems or directories are on any of the slices from slice 3 through slice 7.

If the `/opt`, `/proc`, `/tmp swap` or `/export/home` directories are mounted over any of the slices from slice 3 through slice 7, reinstall Solaris 9 using the instructions from Chapter 2 before you follow any of the remaining lessons in this chapter.

The following lessons in this chapter require the test system to have a specific disk layout. This lesson shows the user how to set up the system's hard drives for future lessons. As mentioned above, if your system only has one hard drive, is highly advisable that you install the operating system as shown in Chapter 2. It is possible to perform following lesson with only one hard drive, but use caution not to overload the RAID volumes with a big file, or the read/write operations could severely lag the system.

Understand that once the system disk (the disk that holds the operating system) has been created, it can not be reconfigured again. Once the slices on the system disk are set, they can not be changed. The only things a system administrator can do to change the slices are to reinstall the operating system to change the system disk's slice layout, or reformat the disk and recover the operating system from a backup copy.

In this lesson, readers will use the **format** command to reformat the second hard drive on the test workstation. After the proper slices have been created, the **newfs** command will be used to create a UFS file system on the slices.

If the test workstation does not have a second hard drive:

- Do not use the **format** command:
- Skip all the **format** commands and start the lesson on step 26.

These slices will be used in future examples to demonstrate the different types of volumes.

1. Log in as the root user.

2. Open a Terminal window.
3. For systems with only one disk, skip down to step 26. For systems with two disks, continue with the next step.
4. Type the command **mount**

*The **mount** command without any arguments shows the currently mounted partitions.*

5. Look at the output from the mount command. Make sure that *none* of the slices on the *second* hard drive are mounted.

If necessary, type the command:

umount < name of the mount point >

for each mounted slice on the second hard drive.

6. Type the command **format**

*The **format** command displays information about two hard drives on the system, as shown below. Your display will look somewhat different.*

format

searching for disks... done

AVAILABLE DISK SELECTIONS:

0. **c0t0d0** < WDC AC25100L cyl 10670 alt 2 hd 15 sec 63>

/pci@1f,0/ide@d/dad@0,0

1. **c0t2d0** <ST315320A cyl 29649 alt 2 hd 16 sec 63> **disk2**

/pci@1f,0/ide@d/dad@2,0

Specify disk (enter its number):

*As shown above, two hard drives should be shown with the **format** command.*

7. After the **Specify disk** prompt, type the number (0 or 1) that appears at the left of the listing for the disk that you want to format.

*On most systems, the operating system is on the **c0t0d0** hard drive. If a message appears saying **Warning: Current Disk has mounted partitions**, that could indicate that the selected disk is the disk that hold the operating system.*

*If you have selected the hard drive that contains the operating system, type the command **disk** from the **Format>** prompt. This will show the disk selection menu again. Select the other hard drive.*

8. Type the command **partition**
9. Type the command **print**
Take special note of the size of the hard drive. Disk slice 2 represents the entire disk, so the size of slice 2 is the size of the hard drive. If an 800 MB hard drive is installed, disk slice 2 would be 800 MB in size, as shown on the partition table.
10. Type the command **modify**
11. Select menu item **1) All Free Hog**
12. Type **Yes** to the question **Do you wish to continue creating a new partition table based on above table [yes]?**
13. Type **7** to choose the free hog partition.
The free hog partition "hogs up" all the remaining free space if any is left after the other slices have been created.
14. At the prompt **Enter size of partition '0' [0b, 0c, 0.00mb, 0.00gb]:** press the Return key, so that the size is 0.00 mb.
15. At the prompt **Enter size of partition '1' [0b, 0c, 0.00mb, 0.00gb]:** press the Return key, so that the size is 0.00 mb.
16. At the prompt **Enter size of partition '3' [0b, 0c, 0.00mb, 0.00gb]:** type in 1/4 of the hard drive's total space.

For example, if the hard drive is 800 MB, you would type in **200mb**. If the hard drive is 4 GB, you would type in **2.0gb** on the command line.

17. At the prompt **Enter size of partition '4' [0b, 0c, 0.00mb, 0.00gb]:** type in 1/4 of the hard drive's total space.
18. At the prompt **Enter size of partition '5' [0b, 0c, 0.00mb, 0.00gb]:** type in 1/4 of the hard drive's total space.
19. At the prompt **Enter size of partition '6' [0b, 0c, 0.00mb, 0.00gb]:** type in 1/4 of the hard drive's total space. *Slice 7 automatically takes up any remaining space because it is the Free slice.*
20. You'll see a listing of the partitions that you have set up. At the end of it, the you will be asked:
Okay to make this the current partition table [yes]?
Type **yes**
21. When asked to enter the table name, you can type whatever you want (*it will not really be used for this lesson*).
*Here, type in **dog***
22. When asked if the disk should be labeled, type **yes**
23. At the **partition>** prompt, type **print**
The print command shows the partition created on the hard drive.
24. At the **partition>** prompt, type **quit**
25. At the **format>** prompt, type **quit**
26. (*Start the lesson here if you have only one hard drive. Continue here if you have two drives.*)

Type the command **newfs /dev/rdisk/c#t#d#s3** (where **c#t#d#** is the device name of the fourth slice on the drive that you have chosen).

The commands shown in this step, and in the steps that follow, are the typical ones for a standard setup. Depending on your hardware setup, these numbers may be different. If so, substitute those numbers in the commands shown below.:

- For systems with **two** hard drives, this is usually **newfs /dev/rdisk/c0t2d0s3**
- For systems with only **one** hard drive, this is usually **newfs /dev/rdisk/c0t0d0s3**

At the prompt that begins **newfs: construct a new file system....** type **y**
This creates a new file system on the fourth slice of the drive that you have chosen..

27. Type the command **newfs /dev/rdisk/c#t#d#s4** where **c#t#d#** is the device name of the fifth slice on the drive that you have chosen.
 - For systems with **two** hard drives, type **newfs /dev/rdisk/c0t2d0s4**
 - For systems with only **one** hard drive, type **newfs /dev/rdisk/c0t0d0s4**

At the prompt that begins **newfs: construct a new file system....** type **y**

28. Type the command **newfs /dev/rdisk/c#t#d#s5** where **c#t#d#** is the device name of the sixth slice on the drive that you have chosen.
 - For systems with **two** hard drives, type **newfs /dev/rdisk/c0t2d0s5**
 - For systems with only **one** hard drive, type **newfs /dev/rdisk/c0t0d0s5**

At the prompt that begins **newfs: construct a new file system....** type **y**

29. Type the command **newfs /dev/rdisk/c#t#d#s6** where **c#t#d#** is the device name of the seventh slice on the drive that you have chosen.
 - For systems with **two** hard drives, type **newfs /dev/rdisk/c0t2d0s6**
 - For systems with only **one** hard drive, type **newfs /dev/rdisk/c0t0d0s6**

At the prompt that begins **newfs: construct a new file system....** type **y**

30. If the eighth slice (slice 7) has any reasonable amount of disk space (greater than 50 MB), type the command shown below.

- For systems with **two** hard drives, type **newfs /dev/rdisk/c0t2d0s7**
- For systems with only **one** hard drive, type **newfs /dev/rdisk/c0t0d0s7**

At the prompt that begins **newfs: construct a new file system....** type **y**

Lesson 26.6 Creating a RAID 0 (Concatenated) Volume

In this lesson a RAID 0 volume will be created. Under normal circumstances, a RAID volume is always created on a *different* drive than the drive being mirrored. However, for teaching purposes, this lesson also shows how to create a RAID volume on a one-drive system. In real life, you would never do this.

WARNING

- If your system has only *one hard drive*, and you did *not* install Solaris 9 according to the instructions in Chapter 2:
 - Make sure that no critical file systems or directories are on any of the slices from slice 3 through slice 7.
 - If the `/opt`, `/proc`, `/tmp swap` or `/export/home` directories are mounted over any of the slices from slice 3 through slice 7, reinstall Solaris 9 using the instructions from Chapter 2 before you follow any of the remaining lessons in this chapter.

1. Log in as the root user.
2. If your system only has *one* hard drive, follow this step.
 - a. In the `/etc/vfstab` file, delete any line that references slice3, slice4, slice5, slice6 or slice7.

For example, the following lines were deleted on the author's SunBlade 100:

```
/dev/dsk/c0t0d0s3 /dev/rdisk/c0t0d0s3 /slice3 ufs 2 yes -  
/dev/dsk/c0t0d0s4 /dev/rdisk/c0t0d0s4 /slice4 ufs 2 yes -  
/dev/dsk/c0t0d0s5 /dev/rdisk/c0t0d0s5 /slice5 ufs 2 yes -  
/dev/dsk/c0t0d0s6 /dev/rdisk/c0t0d0s6 /slice6 ufs 2 yes -
```

- b. Type the following commands:
umount /slice3
umount /slice4
umount /slice5
umount /slice6

3. Open the SMC.
4. Left double click on the "This Computer" icon in the left pane.

 This Computer (sun100)

You can also left click on the Turner icon (the blue icon to the left of the "This Computer" icon).

5. Left double click on the "Storage" icon in the left pane.

 Storage

A popup window might appear with the title "Log in: User Name."

This window asks for an authorized SMC user. Type in the username **root** and the root user's password (previous chapters have set up the root user's password as **root123**).

6. Left double click on the "Enhanced Storage" icon in the left pane.

 Enhanced Storage

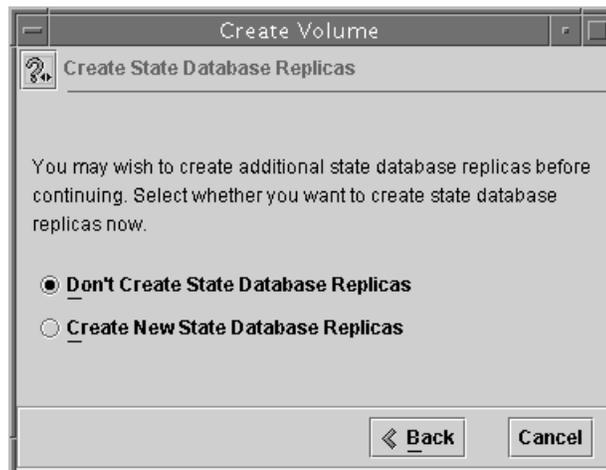


7. Highlight the Volumes icon.

8. Left click on Action in the Menu bar.

9. Left click on Create Volume...

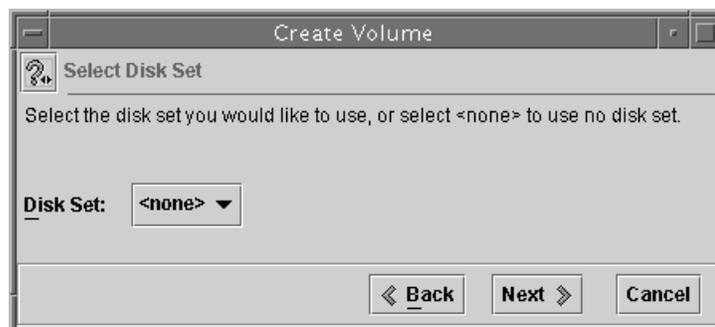
A popup window should appear, titled "Create Volume"



The first screen lets you choose either **Create New State Database Replicas** or **Don't Create State Database Replicas**. (State Database Replicas should already exist from the previous lesson.)

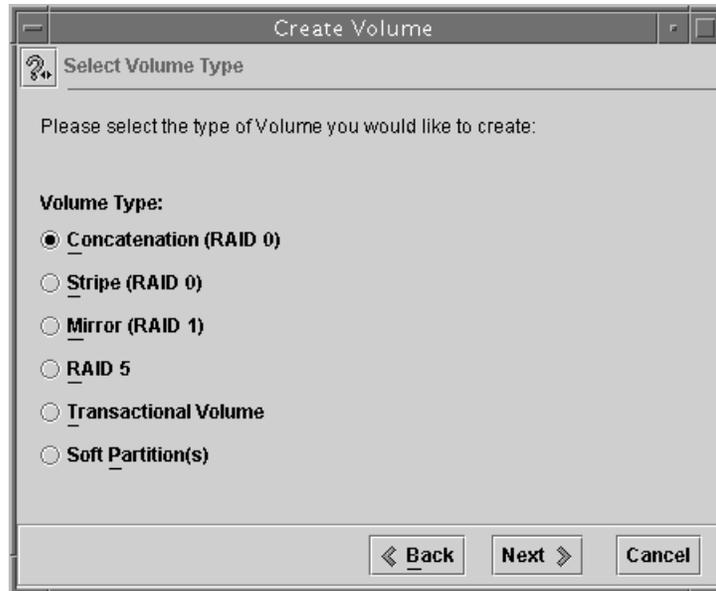
10. Select the option (*) **Don't Create State Database Replicas**. Then left click on the Next > button

The next screen asks what disk set to use.

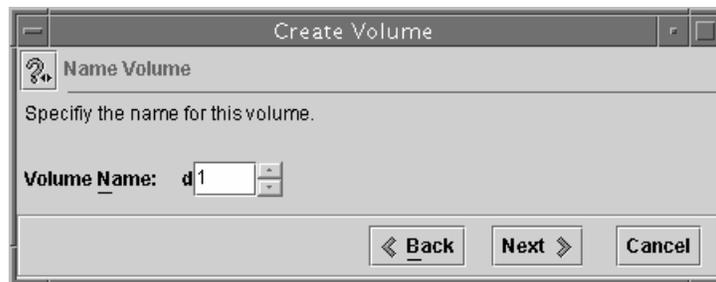


11. The only option should be **<none>**. Make sure that **<none>** is selected, then left click on the Next > button

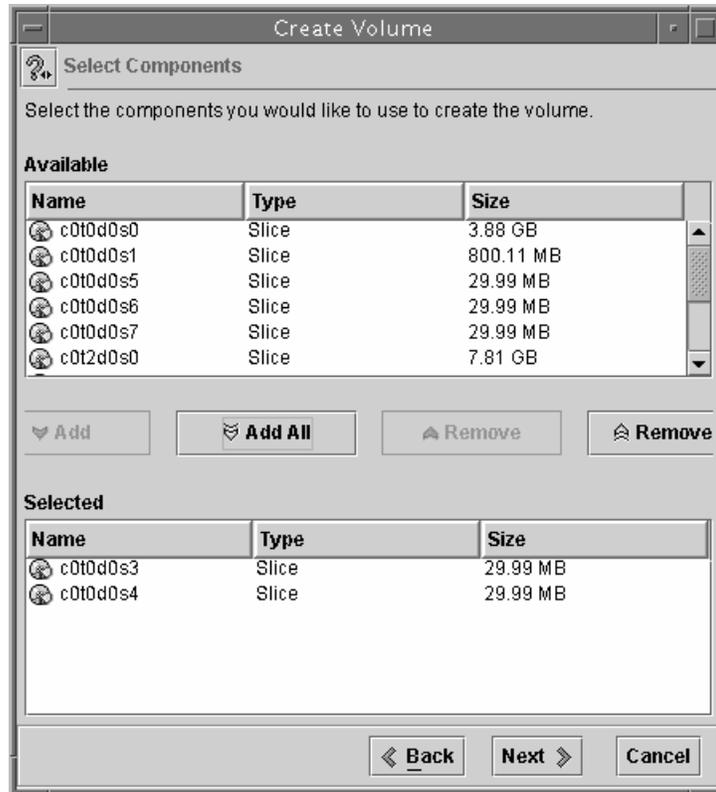
The next screen asks what type of RAID device to create.



12. Select Concatenation (RAID 0) from the menu, then left click on the Next > button.
Now it's time to create the metadvice name.



13. Select the Volume Name **d1**, then left click on the Next > button.
This screen lets the administrator choose the slices used for the RAID 0 device.



14. Choose the appropriate slices for your system:

- For systems with only one hard drive, select slice 3 and slice 4 (typical values are **c0t0d0s3** and **c0t0d0s4**), then left click on the Next button. The order of the slices does not matter.
- For systems with two hard drives, select slice 3 and slice 4 on the second hard drive (typical values for slice 3 and 4 on a second hard drive are **c0t2d0s3** and **c0t2d0s4**), then left click on the Next button. The order of the slices does not matter.

For some reason, SVM will sometimes not recognize a new partition on a disk that has just been formatted. If the second hard drive's slices s3, s4, s5 and s6 don't seem right:

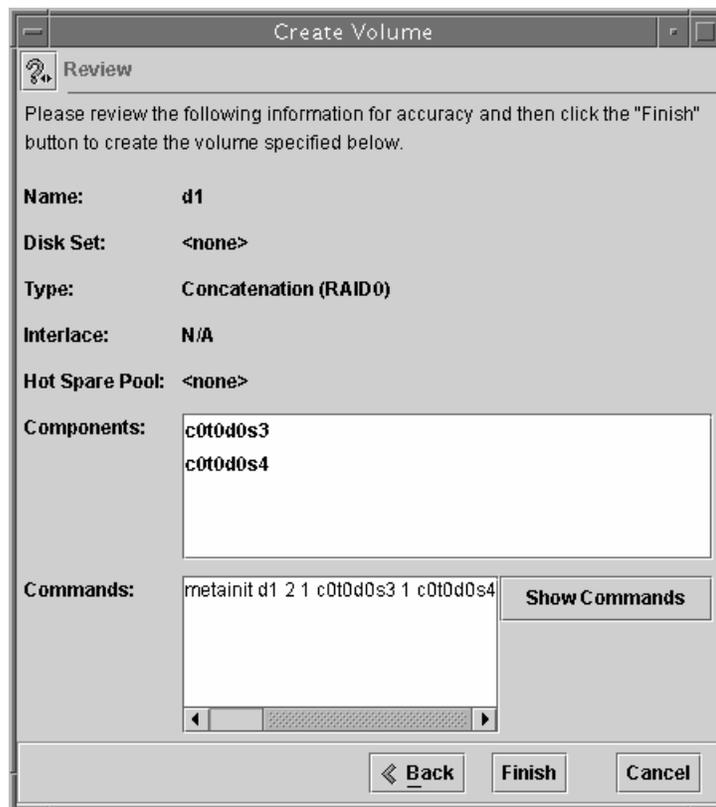
- Quit the "Create Volume" wizard and double click on the hard drives through the Disks icon located under the Storage icon.*
- Re-run the "Create Volume" wizard*
- If the second drive still does not seem right, remove all the State Database Replicas and create new State Database Replicas.*

The next screen gives the administrator the option of creating a Hot Spare Pool.



15. Select (*) No Hot Spare Pool

*In the Commands field at the bottom, the Review screen shows the **metainit** command that will be used to create the volume.*



16. Left click on Finish.
17. Highlight the Volumes icon in the left-hand pane.
*There should be a volume named **d1** in the right pane. If for some reason the **d1** volume is not displayed in the right pane, left click on View, then on Refresh.*
18. Left click on Console
19. Left click on Exit

Lesson 26.7 Mounting a Volume

In this lesson a Volume will be mounted, so that it can be used. For some reason, the SMC and the SVM do not create mount points and do not mount the volumes that have been created. As strange as it may seem, the SMC will modify locally mounted hard drives and volumes, but will not mount volumes.

To mount a volume, run the following commands outside of the SVM:

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command `cd /`
4. Type the command `mkdir /dlmount`
5. Type the command `newfs /dev/md/rdisk/d1`
6. The question `newfs: construct a new file system /dev/md/rdisk/d1: (y/n)?` will appear on the screen. Type `y`
7. Type the command `fsck /dev/md/rdisk/d1`
The `fsck` command is used to check a file system for problems. In this case, the file system most likely will not have problems, because it was just created. One key point that should be understood is that the volume acts just like a regular file system except it has the `/dev/md` reference.
8. Type the command `mount /dev/md/dsk/d1 /dlmount`
The `d1` volume was created in the previous lesson.
9. Type the command `cd /dlmount`
10. Type the command `mkfile 2m 2mbfile`
The command `mkfile 2mbfile` creates a 2 MB file.
11. Type the command `cp /etc/vfstab /etc/vfstab.backup`
This copies the `/etc/vfstab` file in case something goes wrong.
12. In `vi` or another text editor, add this line to the last line of the `/etc/vfstab` file:
`/dev/md/dsk/d1 /dev/md/rdisk/d1 /dlmount ufs 1 yes -`
13. Save the file and exit.
14. Type the command `umount /dlmount`
15. Type the command `mount /dlmount`
The command `mount /dlmount` reads the `/etc/vfstab` file for the mount point. This is a good check to run before rebooting the system. If the file system can not be mounted through the entry in the `/etc/vfstab` file, it will be shown with the `mount` command. If there is a mistake in the `/etc/vfstab` file, Solaris 9 could halt on a reboot. This is a good sanity check to make sure that the operating system will reboot without a `vfsab` halt.
16. Type the command `init 6`
The command `init 6` is used to gracefully reboot the workstation.
17. After the system reboots, log in as the root user.
18. Open a Terminal window.
19. Type the command `cd /dlmount`
20. Type the command `ls`
The file `2mbfile`, which was created earlier, should be listed in the directory.
21. Type the command `mount | grep /dlmount`
The command uses the pipe symbol (`|`) between the `mount` command and the `grep` command. The pipe (`|`) symbol can be found below the Enter key on a PC keyboard, or above the backspace key on a Sun keyboard. This command shows that the `/dlmount` directory is mounted.
22. Type the command `df -h`
The command `df -h` shows all the space on the mounted file systems. The `df -h` option is a new feature in Solaris 9. Previous versions of Solaris used the `df -k` command to show the size of the hard drive in kilobytes. The `-h` option shows the size of the file system in gigabytes or megabytes.

Lesson 26.8 Changing a Mount Option in the SMC

In this lesson readers will change the way the **d1** volume is mounted. Unfortunately, there are some versions of SMC 2.1 that do not work correctly when it comes to mounting and un-mounting, and to using some of the windows associated with the Mount Icon. Don't be too surprised if the **mount** options for the **d1** volume can not be changed by the SMC.

1. Log in as the root user.
2. Open the SMC.
3. Left double click on the "This Computer" icon in the left pane.

 This Computer (sun100)

You can also left click on the Turner icon (the blue icon to the left of the "This Computer" icon).

4. Left double click on the "Storage" icon in the left pane.

 Storage

A popup window might appear with the title "Log in: User Name."

*This window asks for an authorized SMC user. Type in the username **root** and the root user's password (previous chapters have set up the root user's password as **root123**).*

5. Left double click on the "Enhanced Storage" icon in the left pane.

 Enhanced Storage

6. Left click on the icon labeled "Mounts and



Shares."  Mounts and Shares

7. Highlight the Mounts Icon. 

*The right view pane should show the **/d1mount** mount. If it does not, left click on View, then left click on Refresh.*

8. Highlight the **/d1mount** mount point 

9. Click on the Properties button located on the Tool bar. 

The Properties Icon is located on the top center of the screen. It looks like a blue piece of paper with a checkmark on the lower right corner.

10. A window should appear with the title "Mount Properties for **/dev/md/dsk/d1**"

Left click on the **Mount** tab.

11. Make sure there is a check on the box **Mount Directory At Boot**.

12. Left click on the **Advanced** tab.

13. In the Handling a Damaged File System box, click on (*) **U**nmount the file system

*This will unmount the **d1** volume if it becomes damaged, instead of halting the operating system from booting.*

14. Left click on the **OK** button.

15. A warning screen may appear stating that the mounted resource will be un-mounted and then re-mounted

If this happens:

- a. Click on **Update**.
- b. If nothing happens, click on **Cancel**.

This is the part where some versions of SMC will not work properly. Nothing may happen, no matter what button you click on.

- c. If neither of these buttons does anything, left double click on the Window Menu button.  (The Window Menu button looks like a dash (-) inside a square box located on the top of the window.)

16. Left click on **C**onsole in the Menu bar.

17. Left click on **E**xit.

Lesson 26.9 Adding Extra Space to the Concatenated Volume

In this lesson readers will add an extra slice to the **d1** volume that was created earlier. The concatenated volume can add extra slices. It is a good idea to make sure no user is using the volume during this operation.

There are times that the new volume will not recognize the underlying slices and the increased size. One sure fire method of making sure that the volume has grown to its full size is to back up the data to a tape or to another slice, then use the **newfs** command to create a new file system.

1. Log in as the root user.
2. Open the SMC.
3. Left double click on the "This Computer" icon in the left pane.

 This Computer (sun100)

You can also left click on the Turner icon (the blue icon to the left of the "This Computer" icon).

4. Left double click on the "Storage" icon in the left pane.

 Storage

A popup window might appear with the title "Log in: User Name."

*This window asks for an authorized SMC user. Type in the username **root** and the root user's password (previous chapters have set up the root user's password as **root123**).*

5. Left double click on the "Enhanced Storage" icon in the left pane.

 Enhanced Storage

6. Left click on the Volumes icon.

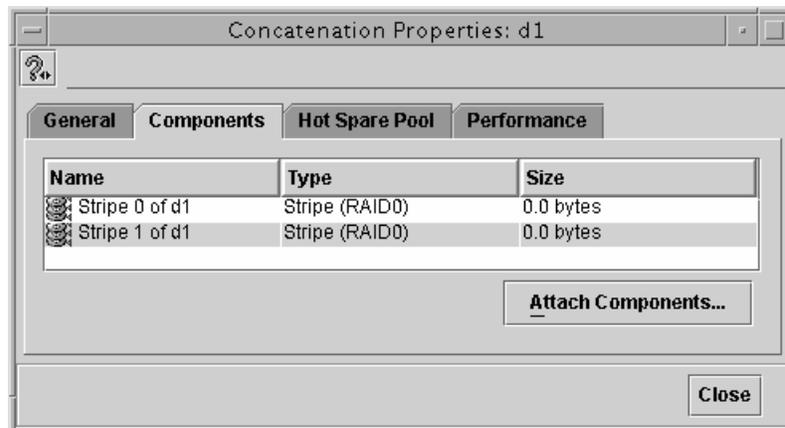


7. Right click on the **d1** volume.

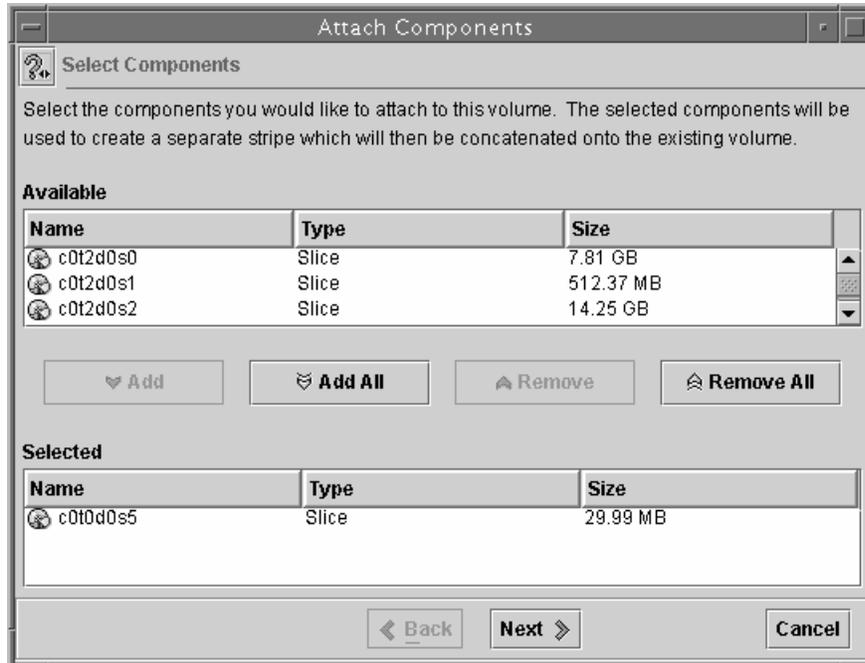
8. Left click on the Properties item



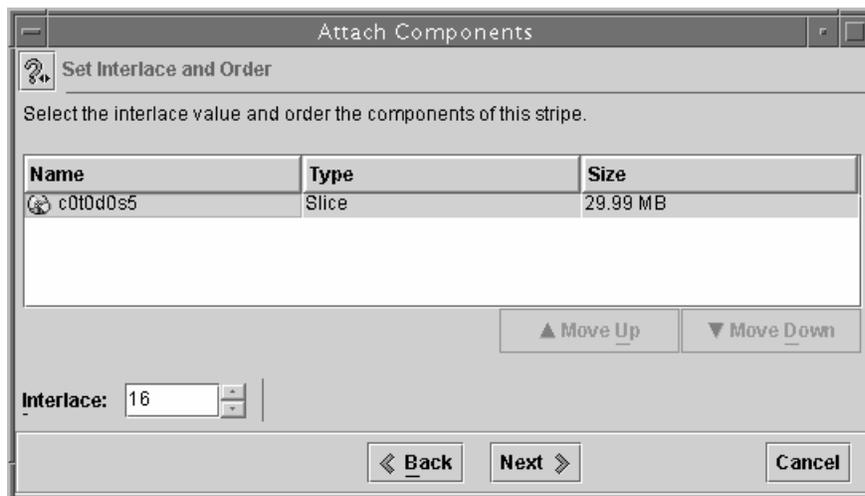
A window should appear with the title "Concatenation Properties: d1"



9. Left click on the Components tab.
10. Left click on the Attach Components... button located at the bottom of the window.
11. Select the **c0t2d0s5** slice (the numbers **c0t2d0** could be different, depending on your hardware).



12. Left click on the Add button.
13. Left click on the Next > button.



14. To accept the default interface size of 16 KB, left click on the Next > button.
The final window is a confirmation window. Notice the Commands field at the bottom of the window. Write down these commands so that someday if the GUI is not available, the same commands could be used from a telnet session to create the volume. In essence, learn from the SMC how to do things in the command line.
15. Left click on the Finish button.
16. Left click on the Close button.
17. Left click on Console.

18. Left click on Exit.
19. Open a Terminal window.
20. Type the command **df -h**
Notice that the size of the volume is larger now?
21. *There are times that for some reason, the new file system does not understand that extra slices have been added to grow the file system. If for some reason the disk slice attached a device but did not grow, follow these steps:*

WARNING

*The **newfs** command below will destroy all files on the **d1** volume.*

*Before typing the **newfs** command, make sure that you have backed up all critical files on the **d1** volume.*

- a. In a production environment, make sure to save the data to backup media.
- b. Type the command **umount /dlmount**
- c. Type the command **newfs /dev/md/rdisk/d1**
- d. When prompted, type **y**.
- e. Type the command **mount /dev/md/dsk/d1 /dlmount**
This command remounts the /dev/md/dsk/d1 volume to the /dlmount directory.

Lesson 26.10 Deleting a RAID 0 Volume

In this lesson the **d1** volume will be removed, using the SMC. Deleting a volume is a combination of using the SMC and a Terminal window. The **/etc/vfstab** file will also be modified so that it does not try to **mount** the **d1** volume on startup. If the **/etc/vfstab** file is not modified in this way, the next time Solaris 9 reboots, the system will hang.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command **umount /dlmount**
*The command **umount /dlmount** is used to unmount the /dev/md/dsk/d1 volume.*

*If an error message appears saying **umount: /dlmount busy** this indicates that the /dlmount directory is being accessed by a user or a program.*

If this happens:

- a. *Close down the SMC and check to see if any other Terminal window has /dlmount as the current working directory.*
 - b. *If a Terminal window has /dlmount as the current working directory, type the command **cd /** to make sure that the /dlmount directory is not mounted in this Terminal window.*
 - c. *Close down all Terminal windows other than the window that you opened in step 2.*
4. Type the command **cat /etc/vfstab.backup**
This file should have been created earlier in this chapter.
 5. If **/etc/vfstab.backup** exists and does **not** contain a reference to the **d1** volume, type the command **cp /etc/vfstab.backup /etc/vfstab**
 6. If **/etc/vfstab.backup** does **not** exist, very carefully edit the **/etc/vfstab** and remove the line that says:

```
/dev/md/dsk/d1 /dev/md/rdisk/d1/dlmount    ufs    1    yes    -
```

At this point in time the **d1** volume still exists but it is not being used.

7. Open the SMC.
8. Left double click on the "This Computer" icon in the left pane.

 This Computer (sun100)

You can also left click on the Turner icon (the blue icon to the left of the "This Computer" icon).

9. Left double click on the "Storage" icon in the left pane.

 Storage

A popup window might appear with the title "Log in: User Name."

This window asks for an authorized SMC user. Type in the username **root** and the root user's password (previous chapters have set up the root user's password as **root123**).

10. Left double click on the "Enhanced Storage" icon in the left pane.

 Enhanced Storage

11. Left click on the Volumes Icon



Right click on the **d1** volume

12. Right click on the **d1** Icon in the right-hand view pane

13. Select the Delete option.

A warning message will appear with the notification

"Are you sure you want to delete the selected volume(s)?"

Command to be executed **/usr/sbin/metaclear d1**

If a warning message appears saying that "d1 is open", type the command

umount /dlmount again.

14. left click on the Delete button

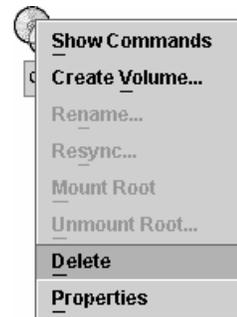
15. Click on View

16. Click on Refresh

At this point in time there should be no more volumes

17. Left click on Console

18. Left click on Exit



Creating a RAID 1 Volume

A RAID 1 volume is typically known as a mirror. With a RAID 1 volume two hard drives simultaneously write the same information down at the same time. During read operations the two hard drive will both read information. RAID 1 volumes can be created for the root (/) partition or other partitions.

Lesson 26.11 Creating a RAID 1 Volume in the SMC

In this lesson a RAID 1 volume is created. Remember, a RAID 1 volume is a mirror. With a Mirror two different slices are reading and writing at the same time. If one device becomes disabled, the second device will continue to work. This is a very popular option with operating systems.

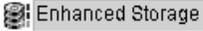
To create a RAID 1 volume, two RAID 0 volumes must be created. The two RAID 0 volumes are then mirrored to each other through a RAID 1 mirror. If either of the RAID 0 devices are damaged, the RAID 1 device will continue to work.

d3 will be a RAID 0 device over the slice c0t2d0s3

d4 will be a RAID 0 device over the slice c0t2d0s4

d10=mirror created over d3 and d4

1. Login as the root user.
2. Open the SMC.
3. After the initial splash screen disappears, left click on the icon labeled "This Computer"
4. Left double click on the icon labeled "Storage." 

- Left double click on the icon labeled “Enhanced Storage.”  Enhanced Storage
If a window appears with the title “Log In: User Name,” type in the root user’s name (**root**) and the root user’s password.
- Left click on the Volumes Icon.



Two RAID 0 devices (concatenated devices) must be produced before a RAID 1 device can be produced.

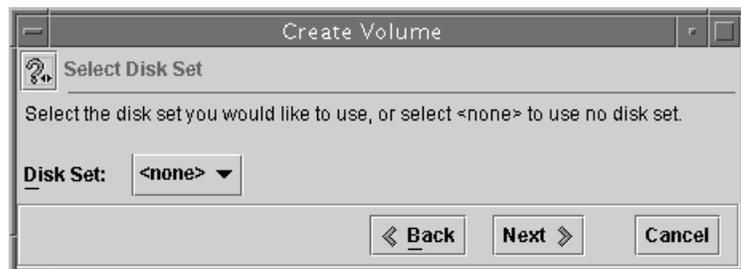
- Create two RAID 0 Concatenated Volumes:
d1 (c0t0d0s3)
d2 (c0t0d0s4)

Lesson 26.6 demonstrated how to make RAID 0 Concatenated Volumes.

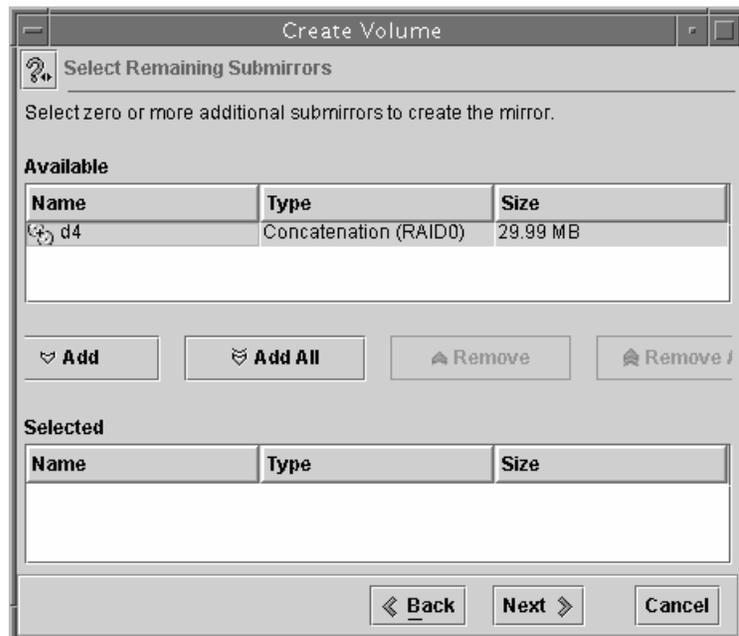
- Left click on Action in the menu bar.
- Select the option Create Volume...
- Select the option Don’t Create State Database Replicas



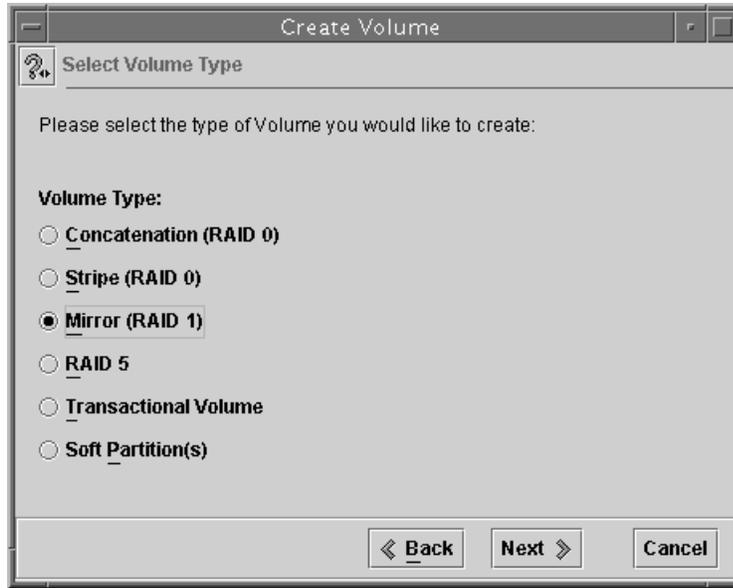
- Left click on the Next > button.
- For the Disk Set option, select: <None>
- Left click on the Next > button.



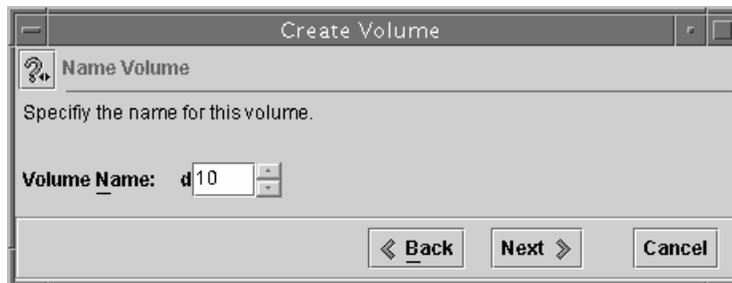
- Select the d4 RAID device.



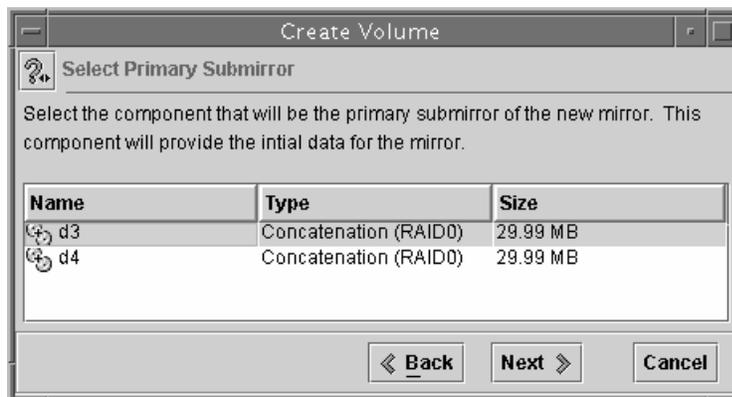
15. For the Volume Type option, select **Mirror (RAID 1)**
16. Left click on the Next > button.



17. For the Volume Name option, specify: **d10**
18. Left click on the Next > button.



19. In the Select Primary Submirror window, select only **d3** as the primary mirror. Do not select **d4**.
20. Left click on the Next > button.



21. In the Select Remaining Submirrors window, select **d4**.

22. Left click on the Add > button.
23. Left click on the Next > button.
24. For Read Option, select: Round Robin.
25. For Write Option, select: Parallel.
26. For Pass Number, select: 1.
27. Left click on the Next > button
28. Left click on the Finish > button.

As shown in the Commands field, the commands used to create the mirror are:

```
/usr/sbin/metainit d10 -m d1  
/usr/sbin/metattach d10 d2
```

It can take up to 10 minutes to create a mirror, even if the mirror is very small. The new mirror has to be created and then synchronized.

29. Left click on View
30. Left click on Refresh
SMC is not too good about refreshing the view when something changes. Use the View, Refresh option if something does not seem right. Use the option several times if necessary. There are times when the Refresh option has to be used several times to see items properly.
31. You should now see a mirror icon for the **d10** mirror
32. Left click on Console.
33. Left click on Exit.



The **d10** mirror now needs to be mounted into a directory. To do this, follow these steps:

34. Type the command **mkdir /d10mount**
35. Type the command **newfs /dev/md/rdsk/d10**
36. Type the command **mount /dev/md/dsk/d10 /d10mount**
37. Type the command **cp /etc/vfstab /etc/vfstab.backup**
38. Add the following line to the **/etc/vfstab** file:
/dev/md/dsk/d10 /dev/md/rdsk/d10 /d10mount ufs 1 yes -
39. Save the file and exit.
40. Type the command **umount /d10mount**
41. Type the command **mount /d10mount**

The mirror should now work.

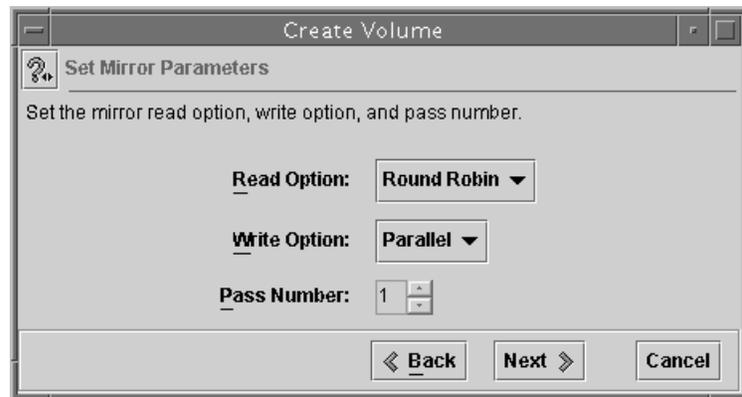
State Database Best Practices

This section describes some advanced techniques for the layout of state databases.

For a system with only one drive, a minimum of three state databases would have to be on that disk. Unfortunately, if the area of the disk that holds the state databases becomes damaged, the virtual file system will become unrecoverable.

This does not necessarily mean that the entire server will die in all cases. For example, if the root file system is not part of the volumes, the root file system will not be damaged. Only the disk slices under a volume will become damaged or unreadable. Try to place several state database replicas on each slice, so that if a single slice dies (with all the state databases on that slice) it will not take down the SVM volumes.

One obvious question might be, why would anyone want to use volume management on a single disk? The answer is simple. This way, a RAID 1 device can be started on a single disk, then when that volume becomes full, another disk can be added to the system. The RAID 1 volume can now be expanded over the second disk without disrupting the server's operations.



If a system has two drives, the author recommends that you use the same number of state databases over both disks. For example, a system could have 9 state databases on one disk and 9 on the other one. If more than two disks are used in a volume, try to spread the databases evenly among the disks and controllers so there is not a single point of failure. Understand that the volume will come up properly after a reboot only if more than 50% of the databases are good.

One very odd feature of SVM state databases is that if a system has two mirrored system disks and one of those two disks dies, the system will halt on a reboot (it can be repaired later).

A simple example can illustrate why this phenomena occurs.

Imagine that a system has two disks. Now imagine that each disk has 3 state databases. In this case the total number of state databases is 6. If one of the disks dies, there would only be 3 state databases out of the original 6 state databases present. Remember, **more than half** of the state databases must be present and in good working condition for SVM to work after a reboot. In this case, because one disk died, there are **only 3 available state databases out of the original 6**, when the minimum number of state databases must be 4 for the SVM to continue running.

In some ways the SVM is a bit odd, because if a system administrator mirrors a disk to achieve higher uptime, but one of the two disks in a mirror dies and the server reboots, the SVM stops. One could say that this just makes the system twice as likely to halt due to a hard disk failure! In that case, why create a mirror in the first place?

In this case, to make the mirrored system robust, the system administrator needs to add a third disk. Even a cheap \$5 disk that holds 100 MB can work as the third disk and can contain state databases. Now, if one of the mirrored disks dies, SVM will keep working, because two remaining disks will have extra state databases to make the total number of good databases greater than 50%.

The same “half of a mirror dies and the reboot halts” phenomenon occurs with disk controllers. If a system has two disk controllers, each with a copy of a mirror, and one disk controller dies, SVM will halt on a reboot. It is a good idea to have three disk controllers on a system to keep the death of one disk or of one disk controller from stopping a clean reboot. The disk and disk controller only need to have the ability to save a state database. Remember, a state database is only 4 MB in size, so even a 100 MB disk would work.

When Solaris 9 starts, if SVM detects a high number of inconsistent databases, the system will stop at run level S (single user mode, otherwise known as the administration run level). At this level, SVM command line tools are the only tools that can be used by the system administrator to repair or delete any bad state databases, so that only good state databases exist. Only when a system has all good, consistent state databases should it continue with the boot process.

During state database maintenance, add extra state database replicas before deleting the original state databases. As mentioned above, a state database replica is the same as a state database once the replication process is finished. These extra state databases are added to ensure that at least three good state databases are in existence at all times. If the original state databases were deleted and the system crashed before new state databases could be created, the virtual file system would be unrecoverable.

If there are too few state databases, they may not be able to handle all read/write requests. This can cause RAID 1 volumes to have I/O problems. So, you should always have at least two databases for each RAID 1 volume. RAID 1 volumes can also have performance problems if there are too many state databases (for example, more than 45) associated with the volume. In this case, the system will spend a too much time trying to keep all the databases updated.

Key Points to Remember

The Sun Volume Manager is a GUI-based tool that lets the system administrator create different types of RAID devices. It is absolutely critical that the reader know how to create volumes and state databases from both the command line and the GUI interface. Remember that if a server's GUI is not running and the SVM state databases need to be repaired, the system will halt at run level S. When the system is in run level S, no GUI is available. Knowing how to use the command line tools can save you a world of hurt!

Chapter 27 IPv6

Lessons in This Chapter

Lesson 27.1 Configurig IPv6 in Solaris 9.....	27-4
Lesson 27.2 Translate Decimal Numbers into Hexadecimal Numbers	27-9

Introduction

In 1981 IPv4 was adopted as the official Internet Protocol. It was thought that IPv4 would be able to handle all the IP addresses required for 40 years after it was created. But now, less than 25 years later, it is no longer sufficient, and IPv6 is starting to become a necessity.

IPv6 will be the eventual replacement for IPv4. Understand that IPv6 and IPv4 can coexist on the same Solaris server without damaging the communications of either protocol stack. When packets travel around the Internet, in theory every Internet host needs a unique IP address. There are some exceptions to this rule. For example, with NAT (Network Address Translation) an ISP (Internet Service Provider) can assign each client a virtual IP address. Because all clients most likely will not connect at the same time, the ISP can get by with fewer IP addresses than it has clients. There are other schemes that also reduce the number of IP addresses that are used on the Internet. Unfortunately, the demand for IP addresses is starting to exceed demand due to the fact there are so many devices out there that can connect to the Internet. Cell phones, PDAs, game stations, all are demanding an Internet address.

IPv4 addresses use what is known as 32 bit addressing. This works out to 4.294 billion possible IP addresses. Unfortunately many of these addresses are not available. These include 127.0.0.1, addresses starting with 0 (0.xxx.xxx.xxx), and addresses starting with 255 (255.xxx.xxx.xxx). So, the actual number of IP addresses available is much less than the theoretical 4.294 billion. Because of this, IPv4 will eventually have to be replaced by an IP address scheme that has more digits. Welcome, IPv6!

IPv6 has some very important improvements over IPv4. IPv6 supports what is known as an *Anycast Address*. This lets a computer determine what route a packet should take. Currently, IPv4 scatters packets over all kinds of routes and devices, so a packet with IPv4 can take a host of different paths to a destination. But because IPv6 packets can be sent over a specific route, traffic that depends on a certain quality grade, such as VOIP (Voice over IP) will be much improved. This means that VOIP transmissions will have less of a choppy sound.

So, when will IPv6 become the Internet-standard protocol? When the pain of routing IPv4 packets becomes so great that companies will have to switch to IPv6. Right now, nobody wants to be the first to implement expensive equipment that can handle IPv6. Although large-scale telecommunications companies can set up the entire IPv6 route and provide all the equipment necessary, smaller companies just don't have the financial incentives for buying IPv4/v6 routers, switches, gateways and for providing staff training. However, Sun has made Solaris 9 IPv6 compatible because Sun has a good tendency of setting up its products ahead of schedule to catch the next big wave.

One very nice feature about IPv6 is that it tends to be a *hands free* protocol, like Microsoft's NetBEUI or Novell's IPX/SPX protocols. The system administrator can set options on IPv6, as desired.

What Does IPv6 Look Like?

A typical IPv6 address will look something like

23b4 : 83c2 : bb3e : 373b : a2df : c293 : eff3 : 2135

As shown above, the 128-bit address is broken down into 8 groups of 4 hexadecimal digits. The first 64 bits deal with network settings, and the last 64 bits deal with the Ethernet address. In IPv6 there is no such thing as a subnet mask. The network and the host portion of the IP address are already set. What would a subnet mask define? Nothing.

If you need a refresher on the hexadecimal (base 16) number system, and on converting among hexadecimal, binary, and decimal, see the "Converting Between Hexadecimal, Binary, and Decimal Numbers" section later in this chapter.

Understanding IPv6 Within the OSI Model

Before a discussion of IPv6 can occur, it is vital that the reader understand the OSI model and where TCP and IP relate in this model. Table 27.1 shows where IPv4 and IPv6 occur in the OSI model. Table 27.1 only shows the part of the OSI model that relates to the TCP and IP protocols.

Layer Name	Function	Examples
Application		
Presentation		
Session		
Transport	Communicate between computers	TCP
Network	Route traffic through the Internet	IPv4 or IPv6
Data		
Physical		

Table 27.1 IPv4 and IPv6 Locations in the OSI model

Understand that the term *TCP/IP* really stands for the TCP protocol that rides inside the IP protocol. TCP packets are embedded inside IP packets. When an application like Telnet, HTTP or FTP sends information to another computer, it sends the information inside a TCP packet. The TCP packet is encapsulated by an IP packet. The IP packet is then moved around the Internet. When the destination computer receives the IP packet, it extracts the TCP packet and reads the data in that packet. When the Internet transitions from IPv4 to IPv6, this will not matter to TCP, because TCP is only being transported inside the IP packet. TCP will not care if it is an IPv4 packet or IPv6 packet.

Both the TCP packet and the IP packet have headers. These headers are completely different components. The TCP packet header does not relate to or interfere with the IP packet header in any way. Figure 27.1 shows an IP packet with a TCP packet inside, including the headers.

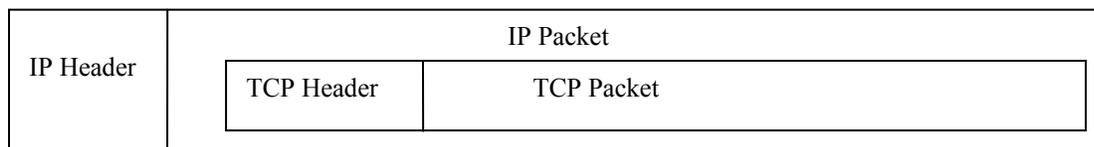


Figure 27.1 IP Packet with TCP Packet Inside

Let's examine the IPv6 header in detail:

First 4 bits	IPv6 version	Which version of IPv6 is being sent.
Next 8 bits	Traffic Class	Class or priority of the packet such as FTP (low priority) or VoIP (high priority.)
Next 20 bits	Flow Label	How the packet flows through routes.
Next 16 bits	Payload Length	Length of the packet.
Next 8 bits	Next Header	What header follows the IPv6 header (TCP header, UDP header, other type of header).
Next 8 bits	Hop Limit	When an IP packet travels between routers, it takes a <i>hop</i> . This is a variable that is used to kill IP packets that are involved in an endless loop between two routers. Each device that handles the packet takes off a hop count. If the packet has a hop count of zero it is dropped. This prevents endless packet travel around a router loop.
Next 128 bits	Source Address	Address of the sender.
Next 128 bits	Destination Address	Address of the receiver.

When IPv6 starts to come out, Internet devices will have to be compliant with what is known as a “Dual-stack” interface. This means that the interface must be smart enough to know when a packet is IPv4 and when a packet is IPv6. Eventually, IPv4 will be phased out when IPv6 takes over.

IPv6 Addressing

Unlike with IPv4, under IPv6, all hosts on the Internet have a unique IP address. There will be no need for NAT and other systems to distribute IP addresses from a pool. Each Ethernet card can be assigned one address, or multiple addresses. Also, each Ethernet card can be assigned different types of addresses. There are three principal types of addresses in IPv6:

Anycast Address	A group identification. Any member of the group can pick up the packet.
Multicast Address	A group identification. All the members of the group pick up the packet.
Unicast Address	Identifies a single interface.

Solaris 9 and IPv6

Solaris 9 fully supports IPv6. For information, see the official Sun Microsystems IPv6 site

<http://www.sun.com/software/solaris/IPv6>

Interface Auto-Configuration

Under IPv6, ethernet cards with IPv6 automatically try to configure themselves, just as they do with the IPX protocol and the MAC protocol. This only occurs on interfaces that have multicast ability. The interfaces take the following steps:

1. A temporary link-local address for the interface is created. This is done by adding the interface's identification to the link-local prefix. If a link-local address is created manually, the address must be tested to make sure that it truly is unique to the local link.
2. The interface then has to make sure that the link-local address is not already being used by another host on the same link. It sends out a message to its neighbors, basically asking “Is this link-local address being used?” If a neighbor sends back a **yes** response, the interface needs to be manually configured.
3. Once an interface determines that it has a unique link-local address, it assigns that address to the interface.

- Now the interface tries to determine the routers in its area. The host sends out a router solicitation to all routers in the “all-routers multicast group.” Routers also periodically send out their own identification messages on the link. The router sends back two pieces of information in the form of two flags.

These flags are:

Managed Address Configuration Flag	Tells the host if it should use stateful auto configuration to obtain its address.
Other Stateful Configuration Flag	Tells the interface if it should use stateful auto configuration to obtain other information.

The routers also send out other prefix information options that are used by stateless address auto configuration to make local and global addresses. Stateful and stateless addresses are processed separately. Because routers constantly send out new router advertisements, the interface updates its information every time it receives a new router advertisement. Other hosts also add information to this message and that is updated by the interface.

- The stateful auto configuration is performed, it then sends back a managed address configuration file.

Lesson 27.1 Configurig IPv6 in Solaris 9

Solaris 9 supports IPv6. In this lesson the reader will configure a workstation with IPv6 so that it can communicate with other systems that have IPv6 enabled.

- Log in as the root user.
- Open a Terminal window.
To open a Terminal window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Tools menu item, then left click on the Terminal window icon.
- Type the command `ifconfig -a`
*This command shows all the Ethernet interfaces on the system. Most Ethernet interfaces have a three-letter identifier such as **hme**, **eri** or **qfe**. The three-letter identifier is followed by a number. The first interface starts with the number zero. The next interface starts with the number 1. So if a system has two 100Mbps Ethernet cards, the names could be **hme0** and **hme1**.*

When the command is run on a system that does not have IPv6 enabled, the output should be similar to that shown below. Write down the interfaces that should work with IPv6 (most likely hme0 or eri0, the first interface). Ignore the interface lo (that is little L little O), which is a dummy interface for the loopback tests..

```
# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
    inet 127.0.0.1 netmask ff000000
eri0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
    inet 192.168.0.4 netmask fffff00 broadcast 192.168.0.255
    ether 0:3:ba:4:c1:3c
```

- Type the command `touch /etc/hostname.<interface name>`
For example `touch /etc/hostname6.hme0`
- Type the command `reboot`
*During the reboot process look at the screen. There should be a message such as **Configuring interface eri0 with IPv6***
- Type the command `ifconfig -a`
The output should look something like the screen shot bellow. Notice the new IPv6 interfaces?

```
# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
    inet 127.0.0.1 netmask ffffffff
eri0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
    inet 192.168.0.4 netmask fffffff0 broadcast 192.168.0.255
    ether 0:3:ba:4:c1:3c
lo0: flags=2000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv6> mtu 8252 index 1
    inet6 ::1/128
eri0: flags=2000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
    ether 0:3:ba:4:c1:3a
    inet6 fe80::203:baff:fe04:c13c/10
```

Other **ifconfig** options to try include:

- To display information only about IPv4, type the command **ifconfig -a4**.
- If a system has more than one card, type the command **ifconfig** by itself.
- To display information only about IPv6 on a specific Ethernet card, type the command **ifconfig <cardname> inet6**.
- In the previous example, the command **ifconfig eri0 inet6** would only show the last line of the display. The command **ifconfig -a inet6** displays IPv6 information on all the cards that have IPv6 enabled.

To assign global and site-local addresses manually, type the commands:

```
ifconfig <cardname> inet6 <global address> up
ifconfig <cardname> inet6 <site-local address> up
```

To configure the interfaces to have static addresses when the system reboots, add the following lines to the `/etc/hostname6.<ethernet_card_namecardname>` file

```
addif <global address> up
addif <site-local address> up
```

Configure Solaris for an IPv6 Network

To configure Solaris 9 to work with an IPv6 network, you must obtain the following key pieces of information:

Local IPv6 address	This is ordinarily handled by the in.npd process automatically. If a specific address is required, this should be provided by the network engineer.
Local IPv6	Tunneling Address This will be an IPv6 address like 3023:219:433:2301:900:203b:fc74:5c9d
Remote IPv6	Tunneling Address This will be an IPv6 address like 3023:219:433:2301:900:203b:fc74:5c9d
Local IPv4 Tunneling Address	For networks that will tunnel IPv6 under IPv4 this should be given by the network engineer or Internet Service Provider.
Remote IPv4 Tunneling Address	This address also needs to be provided by the Internet Service Provider or the network engineer.

Key files used by Solaris when connecting to an IPv6 network

Solaris 9 uses three key files when connecting to an IPv6 network. They are:

- **/etc/hostname6.<interface>**
This is just an empty file. It basically tells Solaris 9 that you want to use IPv6 on this system. The **<interface>** name can be found by using the **ifconfig -a** command. Interfaces have names like **eri0**, **hme0** and **qfe0** depending on the server's hardware.

- **/etc/inet/ipnodes**
This file is used for hostname-to-IPv6 resolution. The contents of the file look something like:

```
2003:214:432:1073:a03:20df:ne43:5c9e data32.sun.com data32
```

- **/etc/hosts.allow**
The **/etc/hosts.allow** file is used with **rarp**, **rsh** and other programs that automatically log into Solaris 9. IPv6 addresses can be added to this file for IPv6 compliance. For example:

```
in.telnetd 210.150.32.108/255.255.255.0 [213:218:433::]/40
in.rlogind 210.150.32.108/255.255.255.0 [213:218:433::]/40
```

The **/etc/inetd.conf** file is used to start the **init** process when an application like **ftp** or **telnet** starts. Modify entries for IPv6 so it looks something like this:

```
telnet stream tcp6 nowait root /usr/local/sbin/tcpd in.telnetd
```

Using IPv6 with DNS

IPv6 is supported with the Solaris 9 DNS server. To configure IPv6 with DNS IPv6, add the AAAA and PTR records.

AAAA records

```
myhostname IN AAAA < IPv6 host address >
```

The PTR record

```
<IPv6 host address> IN PTR myhostname
```

Using the ping Command with IPv6

The **ping** command sends out **ICMP** (Internet Control Management Protocol) packets. With Solaris 9, **ping** sends out **ICMP6** packets. These packets are designed to contact a foreign host with an **ECHO_REQUEST** datagram. If the host has IPv6 and has **ping** enabled, it should respond back with an **ECHO_RESPONSE** datagram reply. If the foreign host sends back a reply, the **ping** command will print the message:

```
<hostname> is alive.
```

If the foreign host does not respond after twenty seconds, the **ping** command will print the message:

```
no answer from <hostname>
```

If the **-s** option is used, an **ECHO_REQUEST** datagram is sent out once a second. If an **ECHO_RESPONSE** datagram is received it will return with the round trip time and packet loss statistics.

Some specific IPv6 ping options are :

```
ping <options> <hostname>
```

```
<options>
```

- a** Ping all addresses.
- A <inet6>** Specify that IPv6 should be tested. Without this option, the default behavior is to test with IPv4 datagrams.
- F <flow-label>** Specify the IPv6 flow label (0 to 1048587) of the packets.

Figure 27.2 shows the output from an IPv6 ping such as `ping -s -A inet6 myhost`:

```
64 bytes from myhost (4::113:b00:20cc:ad4e:21cf): icmp_seq=0. time 4 ms
64 bytes from myhost (fec0::113:b00:20cc:ad4e:21cf): icmp_seq=1. time 4 ms
64 bytes from myhost (4::113:b00:20cc:ad4e:21cf): icmp_seq=2. time 4 ms
64 bytes from myhost (fec0::113:b00:20cc:ad4e:21cf): icmp_seq=3. time 4 ms
64 bytes from myhost (4::113:b00:20cc:ad4e:21cf): icmp_seq=4. time 4 ms
64 bytes from myhost (fec0::113:b00:20cc:ad4e:21cf): icmp_seq=5. time 4 ms
```

Figure 27.2 IPv6 Ping Output

The `ping` command is useful for troubleshooting connections to foreign hosts and for seeing if the foreign host is alive. Understand that some companies have disabled `ping` on their domain sites, servers and routers because hackers in the past have used `ping` in denial of service attacks, where a network device was flooded with ping requests.

Using the `netstat` Command with IPv6

The `netstat` command shows network status information. This command shows what ports are open with each type of protocol (such as TCP or UDP), the status of each interface and some basic network traffic information.

To display some IPv6 network information type the command

```
netstat -af inet6
```

The output of the command is shown in Figure 27.3

UDP: IPv6

Local Address	Remote Address	State	If
.		Unbound	
*.sunrpc		Idle	
.		Unbound	
*.32771		Idle	
*.time		Idle	
*.echo		Idle	
*.discard		Idle	
*.daytime		Idle	

*.chargen	Idle
*.32776	Idle
*.32779	Idle
*.32781	Idle
*.32783	Idle

Figure 27.3 Output from the `ifconfig -af inet6` Command

This output shows the local and remote addresses, port status (in this case the test computer is attached to nothing) and the routing table. Basically the `netstat` command is supported with IPv6.

For information on IPv6 and on the transition from IPv4 to IPv6, see

RFC 1933 An IPv4 to IPv6 transition guide for systems that are being transitioned from IPv4 to IPv6. It is available online at <http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1933.html>.

RFC 1886 This document covers IPv6 DNS. It is available online at <http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1886.html>.

Converting Between Hexadecimal, Binary, and Decimal Numbers

This section discusses how to convert numbers between the hexadecimal, binary and decimal number systems. These are critical skills that a system administrator must know how to perform when dealing with IPv6. The author of this book would like to make this a very strong point.

*“I can not tell you how many times I have converted numbers between decimal, hexadecimal and binary. It is **VERY** important that a system administrator know how to convert these numbers by hand and what they really mean. Imagine running across the hexadecimal number **FF** in a networking document and not knowing what that was referring to?”*

*Steven Beebe
Author*

A successful system administrator or network engineer absolutely needs to be able to translate among hexadecimal, binary and decimal numbers. There are plenty of software programs and calculators that can do this automatically, but if the administrator does not know what is going on in the background, these calculators are of no use.

Hexadecimal, binary and decimal numbers are all used at various places to read and specify settings for a network and an operating system. If you could use a refresher course in converting between these number systems, read this sections that follow.

Hexadecimal Numbers

To thoroughly understand Ipv6 addressing,, it is necessary to understand some basic hexadecimal math. Hexadecimal numbers (often called *hex* numbers) use the base 16 number system. In addition to the decimal digits (0 1 2 3 4 5 6 7 8 9), the hexadecimal digits (A B C D E F) are added.

Figure 27.4 shows what hexadecimal numbers look like.

Decimal Numbers	Hexadecimal Numbers
0	0
1	1
2	2
3	3

4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F

Figure 27.4 Decimal Numbers vs. Hexadecimal Numbers

As shown above, the numbers from 1 to 15 can all be represented in hexadecimal with one digit. Instead of using two digits to represent the number 15 (1 and 5) a single hexadecimal digit is used (F). Instead of using two digits to represent 13, a single hexadecimal digit (D) is used. The first ten hexadecimal numbers (0 through 9) are the same as decimal numbers. When the decimal number 10 is reached, the letter A is substituted for the number 10. This continues until the number 15 is reached, which is substituted with the letter F.

Lesson 27.2 Translate Decimal Numbers into Hexadecimal Numbers

This lesson covers how to translate a hexadecimal number into a decimal number.

1. Translate the decimal number 3 into hexadecimal.
2. Translate the hexadecimal number E into decimal.
Start with A = 10, B = 11, C = 12....
3. Translate the hexadecimal number F into decimal.
4. Translate the decimal number 13 into hexadecimal.

Answer 3 decimal = 3 hexadecimal
 14 decimal = E hexadecimal
 15 decimal = F hexadecimal
 13 decimal = D hexadecimal

5. Translate the number 3f2a into a decimal number

Answer *Convert each hexadecimal digit into its decimal equivalent digit*
 3 hexadecimal = 3 decimal
 f hexadecimal = 15 decimal
 2 hexadecimal = 2 decimal
 a hexadecimal = 10

Create a hexadecimal table with the following values

4096	256	16	1
------	-----	----	---

Place the hexadecimal numbers under the table from right to left

4096	256	16	1
3	15	2	10

Multiply the top row with the bottom row to produce these results

12288	3840	32	10
-------	------	----	----

Add the values together

$$12288 + 3840 + 32 + 10 = 16170$$

- To check out your hex-to-decimal conversion skills, make up a few 4-digit hex numbers and convert them to decimal, using the technique in step 5. To check your answers, use the calculator program in Microsoft Windows (Start, Programs, Accessories, Calculator) or the scientific calculator in Solaris 9 (Workspace menu, Applications, Calculator).

Understanding Binary Numbering

A binary digit can only have one of two values, a one (1) or a zero (0). To understand how many combinations are possible with a binary number, multiply the number 2 x 2 as many times as you have a binary number. If you are given the 3 binary digits, that would be 2 x 2 x 2.

For example

- 1 binary digit has 2 possible combinations
- 2 binary digits = 2 x 2 = 4 possible combinations
- 3 binary digits = 2 x 2 x 2 = 8 possible combinations
- 4 binary digits = 2 x 2 x 2 x 2 = 16 possible combinations

Just to review, the power symbol (^) is used to express the possible combinations shown above. Two raised to the power of zero = 1. After that, the power symbol represents the number 2 multiplied by itself so many times. Figure 27.5 shows the number 2 raised from the power of zero to the power of 7.

$2^0 =$	1
$2 \times 1 = 2^1 =$	2
$2 \times 2 = 2^2 =$	4
$2 \times 2 \times 2 = 2^3 =$	8
$2 \times 2 \times 2 \times 2 = 2^4 =$	16
$2 \times 2 \times 2 \times 2 \times 2 = 2^5 =$	32
$2 \times 2 \times 2 \times 2 \times 2 \times 2 = 2^6 =$	64
$2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 2^7 =$	128

Figure 27.5 Powers of 2, Through 2⁷

In the next section, we will use these powers of 2 to convert numbers from binary to decimal.

Converting Binary Numbers into Decimal Numbers

As you probably know, internally, a computer can not understand a decimal number like 43. Computers only work with binary numbers. Because computers only work with binary numbers, it is critical that readers know how to translate between binary numbers and bits.

Figure 27.6 shows what is known as a binary to decimal conversion chart.

Binary Chart							
128	64	32	16	8	4	2	1
x	x	x	x	x	x	x	x
Decimal Location							

Figure 27.6 Binary to Decimal Conversion Chart

Let's use a binary to decimal conversion chart to convert 01011011 to its decimal equivalent. It's really easy. Write the number as shown below, then for each 1 in the binary number, add the decimal equivalent that is written above it. For 01011011, we have:

	128	64	32	16	8	4	2	1	
	0	1	0	1	1	1	0	1	
The number above one's		64		+16	+8	+4		+1	= 93

Given the binary number 10110110

	128	64	32	16	8	4	2	1	
	1	0	1	1	0	1	1	0	
The number above one's	128		+32	+16		+4	+2		= 182

To convert a decimal number to a binary number, pick the largest possible binary digit first. Subtract that number from the decimal number. Then pick the next largest binary digit possible, continue to pick the largest binary digits possible until the number zero is reached. Use Figure 27.6 to find the binary locations that represent one's.

The best way to demonstrate this is with examples:

Given the decimal number 139

139	The largest binary digit possible is 128
139 - 128 = 11	The largest binary digit possible is 8
11 - 8 = 3	The largest binary digit possible is 2
3 - 2 = 1	The largest binary digit possible is 1
1 - 1 = 0	No more binary digits

Numbers we have : 128 8 2 1

Translate to binary

	128	64	32	16	8	4	2	1	
	1	0	0	0	1	0	1	1	

Important: Notice that the binary equivalents of 64, 32, 16, and 4 did not occur when we translated the number above. A zero must be put in the columns for those numbers in the binary chart, and in the binary result itself.

The binary number is
10001011

Given the decimal number 37

37	The largest binary digit possible is 32
37 - 32 = 5	The largest binary digit possible is 4
5 - 4 = 1	The largest binary digit possible is 1
1 - 1 = 0	No more binary digits

Numbers we have: 32 4 1

Translate to binary

	128	64	32	16	8	4	2	1	
	0	0	1	0	0	1	0	1	

As above, a zero is put in the columns for 128, 64, 16, 8, and 2, which did not occur when we translated the number.

The binary number is
00100101

Converting between Binary and Hexadecimal

Writing 10110110 all day long would become very tedious. Also, there is a strong possibility that the person writing the ones and zeroes, and the person reading them, would miss a digit. Because of this, binary numbers are often converted to hexadecimal. Hex numbers require only 1/4 as many digits as binary numbers, so they are a very convenient way to write down numbers like 00100110 that is not fatiguing and is more accurate.

Each hexadecimal digit represents four binary digits. For example:

Binary Value	Decimal Value	Hexadecimal Value
1011	11	B
0110	6	6

The conversion process, as shown above, is:

1. Break the number into sets of 4 binary digits. If necessary, add zeroes to the *left* of the number so that the number of digits is evenly divisible by 4 (this was not necessary here).

1011 and 0110

2. Convert these binary numbers to their decimal equivalents:

$$\begin{aligned} 1011 \text{ binary} &= (1 \times 8) + (0 \times 4) + (1 \times 2) + (1 \times 1) = 11 \\ 0110 \text{ binary} &= (0 \times 8) + (1 \times 4) + (1 \times 2) + (1 \times 1) = 6 \end{aligned}$$

3. Substitute the hexadecimal equivalents for the decimal equivalents:

B and 6

B6 is the hexadecimal representation of the binary number 10110110.

To convert from hexadecimal to binary, reverse the process:

1. Write each hexadecimal digit as its decimal equivalent:

B6 yields 11 and 6

2. Convert each decimal equivalent to its binary equivalent:

11 and 6 yield 1011 and 0110

3. Concatenate the numbers:

B6 hexadecimal equals 10110110 binary.

Understanding IPv6 Numerically

IPv6 is an address space that has 128 bits. Each bit can have two combinations: a one (1) or a zero (0). Because each bit represents 2 raised to a power, there are 2^{128} possible IPv6 addresses. As mentioned earlier, the IPv4 addressing scheme uses 32 bits, so under this scheme, there are 2^{32} possible IPv4 addresses.

The following table shows how much larger the IPv6 address space is than the IPv4 address space:

Version	Bits	Total Addresses	Approximate Decimal Equivalent
IPv4	32	2 ³²	4.3 x 10 ⁹
IPv6	128	2 ¹²⁸	3.4 x 10 ³⁹

Table 27.2 IP Address Space

Under IPv4, there is a maximum theoretical limit of over 4 billion different IP addresses. It is a commentary on the explosion of the Internet that this number is now considered dangerously small!

Under IPv6, there are almost 10³⁰ (1,000,000,000,000,000,000,000,000,000) times as many possible IP addresses as there are under IPv4. It's easy to see why this number is considered adequate, both in the short term and in the long run!

For convenience, IPv6 addresses, like IPv4 addresses, are usually specified in hexadecimal. This representation is based on the fact that 2¹²⁸ addresses can be represented by *128 binary digits*.

Let's go from there:

1. It takes 128 binary digits to represent 2¹²⁸ addresses.
2. Each hexadecimal digit can represent 4 decimal digits (because 2⁴=16).
3. It therefore takes 32 hexadecimal digits (128/4) to represent all possible IPv6 addresses.

For convenience, these 32 hex digits are usually arranged in 8 groups of 4 digits. This means that an IPv6 address would look something like

23b4 : 83c2 : bb3e : 373b : a2df : c293 : eff3 : 2135

Since each group in an IPv6 address has 4 hex digits, there are 65,536 possible values for each group, as shown below:

4096 256 16 1
f f f f

Each of the 4 hex digits can have 16 (decimal) possible values (0 through 15). The lowest value is **0000**, and the highest is **ffff**, which converts to decimal as follows:

$$(15 \times 4096) + (15 \times 256) + (15 \times 16) + (15 \times 1) = 65,535 \text{ decimal}$$

This means that each group of 4 hex digits can have 65,536 possible values.

Some readers ask the question "Why do I need to know all this hexadecimal and binary stuff?" The answer is that in both IPv4 and IPv6, you need to use hexadecimal and binary numbers to specify items such as IP addresses, and to know how many possible addresses are represented by the *groups* in an IP number.

For example, if your server documentation says that the server supports the first two groups of IPv6 addresses, you'll need to know how large the Name Service database needs to be to service both IPv4 and IPv6 addresses.

Key Points to Remember

This chapter describes some of the fundamentals of IPv6 on Solaris 9. Understand that IPv6 is a protocol that basically does its own housekeeping, just as the NetBEUI and IPX/SPX protocols did in the past. Some nice points about IPv6 are that it can be configured by the system administrator to choose a specific route to travel between two hosts, and that the quality of service between two hosts can also be pre-set by a system engineer.

It is uncertain exactly when IPv6 will take off or what the exact problems will occur when it does. But both for the current IPv4 addressing scheme and for IPv6, you should definitely be up to speed on how to translate between decimal, binary and hex numbers.

Chapter 28 System Accounting

Lessons in This Chapter

Lesson 28.1 Installing the K22acct and S22acct Scripts	28-3
Lesson 28.2 Modifying The root and adm Users' cron Settings.....	28-7
Lesson 28.3 Creating a Daily Report.....	28-13

Introduction

System Accounting is a new feature in Solaris 9. With this feature it is possible to generate daily and monthly reports on the activity of a user. This accounting software records the time the user accessed the system, the number of programs run by the user and the number of TTY connections (terminal connections) that the user had with a particular server. One use for this accounting information might be to bill a customer based on the customer's use pattern. For example, some data farms import finite element models from customers and then number crunch the data for them on a high end Solaris system. The Solaris system accounting software could generate a report indicating the amount of CPU time that each customer used, which could then be used for billing purposes.

The accounting software takes into account such conditions that more basic accounting shell scripts might overlook. These include system reboots, system time/date changes, changes in run levels and times when the accounting software was down. Each user's activity can be broken down into the following three main categories:

- Connection Accounting** Records the amount of time that a user spent on the system. This includes the user's login and logout times, length of time the user connected to the system and how many tty connections (terminal connections) were made.
- Disk Usage** Records how many data blocks a user is using on a file system. Remember, a block is 512 bytes, or half the size of a kilobyte. To convert blocks to kilobytes, divide the number of blocks by two.
- CPU Usage** Records the total amount of CPU usage used for each process. The processes are recorded by UID (User ID) and GroupID (GID). The start time and cumulative time of the process are also recorded. The commands associated with the process are recorded, as well as the memory that the process used.

In a nutshell, the system administrator has full accounting information regarding a user's login and logout time, the disk space used, the CPU time used and the amount of memory that the user's processes have taken. This accounting information can be used for billing purposes for external clients, or it can be used just to record a user's activity. If a "power user" takes up too much system resources, the system administrator can present this information to the user when discussing the user's system activity.

Unfortunately, the System Accounting software is very complex to use. It is all command line based and the commands tend to be very cryptic. The System Accounting software package is a combination of Bourne shell scripts and some compiled C programs. The shell scripts can be modified to meet a company's needs.

Another problems is that the System Accounting software package relies on some binary files to save its information. These binary files can not be viewed by the `vi` editor or any other text editor. If the files need to be changed, command line utilities must be run to convert the binary files into text files. After the text files are edited,

the files must be converted back into binary files. This conversion from binary to text to binary is a tedious process that can easily cause errors.

One of the nice things about System Accounting is that it gives an administrator a “feel for the server.” If a modem bank attached to a server is overused, the System Accounting software will show the number of users attached to a server and the number of TTY connections to the server.

Another nice thing about System Accounting is that it shows cumulatively which resources are being used by which programs. There are other commands, such as **sar** and **sag**, that can show general information about the system, but they do not show what a particular software program is doing over a period of time.

The System Accounting family also includes the **chargfee** utility, a program that can be used to automatically calculate users' fees. The problem with **chargfee** is that it only records the user's Login Name, the UID and the fee that the user should be charged. It does not break the bill down into specific records. Furthermore, the fee is added to the previous fees for a cumulative fee.

Most companies that are going to charge customers have other software like Oracle 9i, or other database products that can produce HTML output and customized reports. The **chargefee** utility is not a good option for customer billing. It is available for use, but its practical application is somewhat questionable.

Setting up System Accounting

To set up System Accounting, two Bourne shell scripts need to be created. These scripts are saved in the **/etc/rc0.d** and **/etc/rc2.d** directories.

Quick Tip

- Understand that the only shell scripts that will be run when the system changes run levels are scripts whose names start with a capital **S** or a capital **K**.
- By convention, scripts with names that start with a capital **S** are used to start a process or a program, and scripts that start with a capital **K** are used to kill processes and programs.
- All scripts are run in alphanumeric order. First the scripts that start with a **K** are run. The first scripts that are run are scripts with numbers following the **K** (**K##**). If there are two scripts with the same **K##** numbers, the trailing letters are used to determine which script should be run first

After the scripts that start with the letter **K** are run, the scripts that start with the letter **S** are run in the same alphanumeric order.

If you want to run the System Accounting scripts at a particular spot in the sequence, you can change their names. The default names are **k22acct** and **s22acct** understand that customizing the name of one of these scripts can cause problems if the script starts before a critical process or program has been started by a preceding script.

The names of the accounting scripts can be changed (for example, **k22acct** can be changed to **k40acct**), but the script should not be moved to another **/etc/rc*.d** directory. There may be program or

process issues with another run level that the system administrator does not know about when changing to a new run level directory.

The accounting scripts, and their run levels, are:

- | | |
|---------------------------------|--|
| <code>/etc/rc0.d/K22acct</code> | This script is run when the system enters run level zero (0). Run level zero (0) is used when the system is brought down. When a command like <code>init 0</code> or <code>shutdown</code> is used, the scripts in the <code>/etc/rc0.d</code> directory are executed in alphanumeric order. |
| <code>/etc/rc2.d/S22acct</code> | This script is run when the system enters run level two. Run level two is for multi-user mode, without network resources being shared on the network. Because this script starts with an S it is used to start a process or a program. |

Lesson 28.1 Installing the K22acct and S22acct Scripts

In this example, the System Accounting software is setup to monitor user11's activity on the system. The first part of the lesson deals with installing the run control scripts `K22acct` and `S22acct`.

1. Log in as the root user.
2. Open a Terminal window.
To open a terminal window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Tools menu item, then left click on the Terminal icon.
3. Type the command `pkginfo | grep SUNWacctr`
4. Type the command `pkginfo | grep SUNWaccu`
These commands make sure that the System Accounting software has been properly installed on the system. If these software packages have not been installed on the system, insert the Solaris 9 Software 1 of 2 CDROM and then use the `pkgadd` command to install the software. The software is in the `/cdrom/cdrom0/Solaris_9/Product/.virtual_packages` directory.
5. Type the command `ln /etc/init.d/acct /etc/rc2.d/S22acct`
The command `ln` is used to create what are known as symbolic links. A symbolic link can be thought of as giving the same file two file names. The second file is a file name that points to a "real file." In this case the hard link is the file `/etc/rc2.d/S22acct` that points back to the file `/etc/init.d/acct`. If the file `/etc/init.d/acct` is modified, the file `/etc/rc2.d/S22acct` shows the same results.
6. Type the command `ln /etc/init.d/acct /etc/rc0.d/K22acct`
This command creates the symbolic link `K22acct` to the file `acct`. The files `S22acct` and `K22acct` point back to the same "real file" `/etc/init.d/acct`. If a change is made to `/etc/init.d/acct` the `S22acct` and `K22acct` files contain the exact same contents.
7. Type the command `cat /etc/init.d/acct`
The listing below shows the default contents of the `/etc/init.d/acct` file.

```
state="$1"

if [ -z "$_INIT_RUN_NPREV" -o -z "$_INIT_PREV_LEVEL" ]; then
    set -- `usr/bin/who -r`
    _INIT_RUN_NPREV="$8"
    _INIT_PREV_LEVEL="$9"
fi

[ $_INIT_RUN_NPREV != 0 ] && exit 0

case "$state" in
'start')
```

```

[ $_INIT_PREV_LEVEL = 2 -o $_INIT_PREV_LEVEL = 3 ] && exit 0
echo 'Starting process accounting'
/usr/lib/acct/startup
;;

'stop')
echo 'Stopping process accounting'
/usr/lib/acct/shutacct
;;

*)
echo "Usage: $0 { start | stop }"
exit 1
;;
esac
exit 0

```

*This script is fairly easy to follow. Understand that the operating system passes the arguments **start** or **stop** (depending on whether the system is starting or shutting down) to each script in the **/etc/rc*.d** directories. The variable **state=\$1** is used to pick up this start/shutdown condition.*

8. Type the command **/etc/rc2.d/S22acct start**
A message should appear that reads:

Starting process accounting

Until it is stopped, System Accounting will be running in the background, keeping track of the statistics mentioned earlier in this chapter.

The **/etc/init.d/acct** file does two things. When the system enters run level 2 or run level 0 the script starts or kills the accounting software with the command **/usr/lib/acct/startup** or **/usr/lib/acct/shutacct**. The second part of the script uses the case flow control command to start or shut down the accounting software with the variable **state**.

Understanding System Accounting Commands

Almost all of the System Accounting commands can be found in the **/usr/lib/acct** directory. There script **/etc/init.d/acct** and the holiday configuration file **/etc/acct/holidays** are in the **/etc** directory. Most of the accounting commands end in the same four letters “acct.” This is an easy way to spot commands dealing with System Accounting. The scripts can be customized to generate company specific reports.

There are three main scripts that are used with System Accounting. These scripts are run from the **adm** user’s script. The **adm** user is an account that is used internally by the Solaris 9 operating system. These scripts can be customized for company-specific requirements.

The three main scripts are:

- | | |
|----------------|--|
| ckpacct | This script makes sure that the /var/adm/pacct log file does not become too large. The default size of the /var/adm/pacct file is 500 blocks. A block is 512 bytes, so this works out to 250 KB. |
| monacct | This script generates monthly system accounting information. It can be modified so that only the information desired is produced. |

runacct This script is in charge of collecting all the raw data, such as the disk usage, CPU usage, logins and logoffs. It can be modified to produce only the information desired. For example, if CPU usage is not an issue, this information can be removed.

These scripts should be run from a **cron** command because System Accounting depends on the scripts being run at precise intervals. For example, the **runacct** command should be run every 24 hours.

The following important commands related to System Accounting are in the **/usr/lib/acct** directory:

chargefee Used in special situations where a user needs to be charged additional fees. For example, if a company charges \$100 for installing an extra hard drive, this fee can be added to the user's total fee schedule with the **chargefee** command. The **chargefee** command supports the following syntax **chargefee <UID> <AMOUNT>**.
For example, **chargefee user11 200**.

acctcon Records login/logoff records into accounting records. This command is used in the background by the front end with System Accounting commands.

dodisk Performs disk accounting functions. This command is used in the background by the front end System Accounting commands.

prctmp Prints out the session record file **/var/adm/acct/nite/ctmp**. This file is created by the **acctcon** command.

fwtmp Converts binary records in the **/var/adm/wtmpx** file to formatted ASCII records. This is useful if you need to change bad records with a text editor.

wtmpfix Fixes errors in the **/var/adm/wtmpx** file related to changes in system dates or other accounting problems.

Commands in the **/usr/lib/acct** directory include:

startup This command is started from the **/etc/rc2.d/S22acct** run time control script or when the system administrator types the command **/etc/init.d/acct start**. It runs the commands **acctwtmp** and **turnacct** and also cleans out following temporary files:

```
/var/adm/acct/sum/wtmp*  
/var/adm/acct/sum/pacct*  
/var/adm/acct/nite/lock*
```

shutacct This command is run from the **/etc/rc0.d/K22acct** script or when the system administrator types the command **/etc/init.d/acct stop**. The **acctwtmp** command is run to record the shutdown in the **/var/adm/wtmpx** file. The command **turnacct off** is run to shut down System Accounting.

turnacct This is a shell script that is called on to turn on the accounting software. The **turnacct** command should not be run by hand. Only the **startup** and **shutacct** commands should start this script.

prtacct This command lets the user view the contents of the total accounting file **/var/adm/acct/sum/tacct<MMDD>**

Files related to System Accounting

<code>/var/adm/utmpx</code>	Used to record user access records.
<code>/var/adm/wtmpx</code>	Used to record login/logoff records. This file also contains some administration information. This file is not a text file. The command <code>fwtmp</code> must be run to read this file.
<code>/var/adm/acct/sum/tacct<MMDD></code>	This is a date-named file. The last four characters of the file name are the date in MMDD format. For example, if the date is August 23, the file name would be <code>taacct0823</code> . This file saves all the accounting information.
<code>/var/adm/fee</code>	Records the fees that should be charged for each user. As mentioned earlier, the charging of fees should be done by more powerful database/advanced accounting software. This file is checked by the <code>runacct</code> command, and the total fees are calculated for the Daily Report.
<code>/etc/acct/holiday</code>	Records holidays or dates that should be charged “Prime Time” or PRIM time. Examples might include Thanksgiving, Christmas, and other holidays.
<code>/var/adm/acct/sum/</code>	The <code>/var/adm/acct/sum</code> directories hold some of the accounting files.
<code>rprrt<MMDD></code>	This is a date-named file. For example, if the date is June 24, the file name would be <code>rprrt0624</code> .
<code>loginlog</code>	This file records user logins.. Do not confuse this file with the <code>/var/adm/loginlog</code> file that is used to record bad login attempts. This file is used to record the following: YY-MM-DDUSERNAME
<code>/var/adm/acct/fiscal</code>	This directory holds System Accounting information that is related to fees charged to users.
<code>/var/adm/acct/nite/</code>	This directory holds some accounting information and log information. These files are used by the <code>runacct</code> command.
<code>active0624</code>	The name of this file varies with the name <code>active<monthday></code> . For example, on June 24 the file name would be <code>active0624</code> . This file is an activity log. This file shows WARNING: and ERROR: messages related to System Accounting.
<code>daycms</code>	Shows the daily commands that were run on the system, and the resources that these commands used.
<code>lastdate</code>	Holds the last date each process was run.
<code>log</code>	Log file of any errors that were encountered by the System Accounting software.

reboots	Shows the total number of system reboots since the accounting software was installed. This shows the reporting period, the total number of reboots and what run level the reboots occurred at.
wtmperror	Error messages from the accounting software.
cms	CThis is a report that shows all the commands that where run during the accounting period.
lineuse	This is basically a summary of the console connections and terminal sessions used to connect to the server.
statefile	The state of the System Accounting files.
fd2log	An error log file generated by the runacct command.
/var/adm/fee	Records the actual feet that a user is being charged, such as \$50/hr.

Lesson 28.2 Modifying The root and adm Users' cron Settings

The **ckpacct**, **monacct** and **runacct** scripts are started from the adm user's **cron** settings. The **cron** command is used to start a program periodically (such as weekly or daily), depending on the preferences set by using the command **crontab -e** to edit the **cron** file. In this case the adm user's **cron** files are modified so that the **ckpacct**, **monacct** and **runacct** scripts will be run on a periodic basis.

The root user's **cron** command runs the **dodisk** command. The **dodisk** command is used in the background for disk statistics.

WARNING

The following lesson requires using a text editor to modify the adm user's and root user's **cron** files. If an error is made, the server could become unstable.

Use extreme caution when editing these files.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command **EDITOR=vi; export EDITOR**
*The system variable **EDITOR** is used to set up the default text editor. If you would rather use a third party editor, set the **EDITOR=** system variable to the command name used to start that program. Use extreme caution when editing **cron** files. If a mistake is made with the adm or root user's **cron** file, the server could become unstable.*
4. Type the command **crontab -e adm**
5. Add the following lines to the end of the adm user's **cron** file:

```
0 * * * * /usr/lib/acct/ckpacct
30 4 * * * /usr/lib/acct/runacct 2> /var/adm/acct/nite/fd2log
```

```
30 7 1 * * /usr/lib/acct/monacct
```

- The first command says to run **ckpacct** every hour on the hour. This command checks the **/var** directory to see if that directory has at least 500 blocks of free space. If less than 500 blocks are free, accounting is turned off. After 500 blocks become free, accounting will be turned on again the next time that **ckpacct** is executed.
- The second command says to execute the **runacct** command at 4:30 A.M. each day.
- The third command says to execute the **monacct** command at 7:30 A.M. on the first day of each month.

You can customize these commands as needed by changing the parameters. For example, to execute **runacct** at 7:00 P.M., in the second line, change **30 4** to **0 19** (the hours are in military time).

6. Exit from the adm user's **cron** file:
 - If you are sure that the lines are correct, save the file (in **vi**, type the command **:wq!**).
 - If you are not sure that the lines are correct, quit without saving (in **vi**, type the command **:q!**) Then, start again at step 4.
7. Type the command **crontab -e root**
This command will edit the root user's **cron** file.
8. Add the line following line to the end of the root user's **cron** file:

```
30 22 * * * /usr/lib/acct/dodisk
```

This line runs the **dodisk** command at 10:30 P.M. every day (using military time).

9. Exit from the root user's **cron** file:
 - If you are sure that the lines are correct, save the file (in **vi**, type the command **:wq!**).
 - If you are not sure that the lines are correct, quit without saving (in **vi**, type the command **:q!**) Then, start again at step 7.
10. Type the command **/etc/init.d/acct stop**
11. Type the command **/etc/init.d/acct start**
These commands stop and start the System Accounting software so that they re-read any configuration information.
Remember that the **/etc/rc2.d/S22acct** and the **/etc/rc0.d/K22acct** files point back to the file **/etc/init.d/acct**. These run control scripts perform the start/stop functions automatically when the system starts.
To stop the System Accounting software, type the command **/etc/init.d/acct stop** as the root user.

The **/etc/acct/holidays** File

The **/etc/acct/holidays** file can be modified to take into account holidays, such as Christmas, Easter and Thanksgiving.

Most companies keep their production servers running on a 24x7 basis every week of the year, with an annual contract. The collection of holiday vs. non-holiday time information might not be important in every environment.

The first line in this file defines the year, the starting hour for prime time (PRIME), and for the starting hour for non-prime time (NPRIME):

```
Year PRIME NPRIME (this line is not part of the file)
2002 0830 1800
```

This means that for the year 2002, PRIME time starts at 8:30 A.M. and NPRIME time starts at 6:30 P.M. For companies that don't have PRIME and NPRIME times, set Prime Start to 0000 and Non-Prime Start to a blank value.

All other lines in this file define the holidays for that year, by date and name. Here are some sample lines:

Date	Name	(this line is not part of the file)
1/1	New Year's Day	
5/27	Memorial Day	
7/4	Independence Day	
9/2	Labor Day	
11/28	Thanksgiving	
12/25	Christmas	

For these lines, start with the day of the holiday, in MM/DD notation. The text after the date is just a label/reminder for people to see; it is not read or used by the accounting system.

Notes: (1) To insert a comment line, start the line with an asterisk (*).

(2) Because some holidays fall on different dates in different years, this file must be updated for each new calendar year.

The System Accounting Reports

The System Accounting software creates a series of reports. These reports are used to display information such as each user's use of disk space, CPU time, memory and connection times. These reports are based on a 24 hour time period.

The Daily Report

The Daily Report is created by the `/usr/lib/acct/runacct` script and the `/usr/lib/acct/prdaily` script. This report should only be run by the adm user's `cron` file, with the time interval set to daily. The `runacct` script should be run after hours because it is used to give a summary of the activity during the day. If this script is run during the day, it can cause problems at the end of a billing cycle. The `runacct` script can be executed during off hours if a company is running jobs on a server and the accounting needs to be performed the next morning.

As mentioned above, the best way to run a Daily Report is to include a line like the following in adm user's `cron` file. This command runs the Daily Report at 4:30 A.M. each day:

```
30 4 * * * /usr/lib/acct/runacct 2> /var/adm/acct/nite/fd2log
```

The `runacct` command produces raw data files that are not human readable. To produce a human-readable file from this data, type the command:

```
/usr/lib/acct/prdaily > /mydir/mydailyreportfile
```

(Make sure that you have full read/write privileges to the directory in which you place the file `mydailyreportfile`.)

Figure 28.1 shows a typical Daily Report.

*Notes: (1) The name "Daily Report" can be a bit misleading. This report actually reports on system activity since the last time the **prdaily** command was run, no matter what that interval is. To make it truly be a Daily Report, use the **cron** file to have it run exactly every 24 hours.*

(2) In this figure, multiple blank lines have been eliminated. Also, many lines just before the "Last Login" section are not shown.

(3) In this figure, labels from ① to ⑥ have been inserted. These are for reference only. They do not appear in the actual reports.

① Sep 2 10:24 2002 DAILY REPORT FOR sun100 Page 1

② from Fri Aug 30 04:35:12 2002
to Mon Sep 2 10:23:22 2002
1 date change
1 runacct
1 acctcon

③ TOTAL DURATION IS 4668 MINUTES

LINE	MINUTES	PERCENT	# SESS	# ON	# OFF
console	4	0	3	3	5
pts/4	0	0	0	0	1
pts/5	5	0	1	1	1
TOTALS	9	--	4	4	7

Sep 2 10:24 2002 DAILY USAGE REPORT FOR sun100 Page 1

④	LOGIN	CPU (MINS)	KCORE-MINS	CONNECT (MINS)	DISK	# OF	# OF	# DISK	FEE
UID	NAME	PRIME	NPRIME	PRIME	NPRIME	BLOCKS	PROCS	SESS	SAMPLES
		0	TOTAL 0	1	1349	12609	6	3	0
0	root	0	1	1345	7729	1	1	0	159 2 0 0
4	adm	0	0	4	278	0	0	0	180 0 0 0
100	user11	0	0	0	4602	5	2	0	113 2 0 0

⑤ Sep 2 10:23 2002 DAILY COMMAND SUMMARY Page 1

COMMAND NAME	NUMBER CMDS	TOTAL COMMAND SUMMARY							
		TOTAL KCOREMIN	TOTAL CPU-MIN	TOTAL REAL-MIN	MEAN SIZE-K	MEAN CPU-MIN	HOG FACTOR	CHARS TRNSFD	BLOCKS READ
TOTALS	452	13962.20	1.44	41756.81	9682.52	0.00	0.00	1912348032	6979
sun	2	8294.40	0.46	2780.90	17992.19	0.23	0.00	256008192	659
java	1	3570.35	0.14	0.38	24880.46	0.14	0.38	3151360	217

⑥ Sep 2 10:23 2002 LAST LOGIN Page 1

00-00-00	adm	00-00-00	root2	00-00-00	sms-osd
00-00-00	bin	00-00-00	smmsp	00-00-00	sms-pcd
00-00-00	daemon	00-00-00	sms-codd	00-00-00	sms-svc
00-00-00	ids	00-00-00	sms-dca	00-00-00	sms-tmd
00-00-00	listen	00-00-00	sms-dsmd	00-00-00	softadder
00-00-00	lp	00-00-00	sms-dxs	00-00-00	ssp
00-00-00	noaccess	00-00-00	sms-efe	00-00-00	sys
00-00-00	nobody	00-00-00	sms-esmd	00-00-00	uucp
00-00-00	nobody4	00-00-00	sms-fomd	02-09-05	root
00-00-00	nuucp	00-00-00	sms-frad	02-09-05	user11

Figure 28.1 Typical Daily Report

A Daily Report has two main sections. The first part of the Daily Report shows overall server statistics like how long the server was in multi-user mode, if the server experienced a shutdown and how many TTY connections were started. The second part of the report gives detailed information on each user's activity during the reporting

period. This section breaks down by user the statistics for use of system resources such as CPU time, prime and non-prime connect time, and disk space.

The first section shows the following information about overall system use:

Note: In the discussion that follows, the symbols such as ① refer to the labels in Figure 28.1.

- ① The first line in the report shows the time that the report was run, the day the report was generated, and the name of the server.

It is a good idea to type the command **date** to view the system's current date and time. If the system's date or time is inaccurate it will create problems for the System Accounting programs. However, if the date or time needs to be changed, do make the change during the accounting cycle. If you do, incorrect accounting data could be generated. If clients use the server from 7:00 A.M. to 7:00 P.M., change the time after 7:00 P.M. Otherwise, time could be inappropriately added to, or subtracted from, the accounting information.

- ② The next two lines show the start and stop date and time of the Daily Report. Figure 28.1 shows one of the reasons that this report should be generated by a **cron** job. This report covers more than 24 hours. By using a **cron** job, you can make sure that report always runs at the exact same time each day.

The top part of the report also shows if there were any problems during the reporting period, such as a server crash, shutdown, reboot, and other problems from the `/var/adm/wtmpx` file or from the **acctwtmp** program.

- ③ The lines in the section that starts with the label "**TOTAL DURATION**" show how much time was used by each terminal connection. The following variables are used in the report:

TOTAL DURATION	How much time was the system in multi-user mode. In other words, how much time the system was up while users logged in.
LINE	The name for each terminal line.
MINUTES	How many minutes each terminal line was open.
PERCENT	The time that each terminal line was used, divided by the total amount of time the system was in multi-user mode. (PERCENT = MINUTES/TOTAL DURATION. In Figure 28.1, these percentages have been rounded down to zero.
# SESS	The total number of times the terminal line was used to start a session.
# ON	Not used any more! Ignore
# OFF	The total number of times that a session ended on each line. This includes user logoffs and times when sessions were interrupted for some reason.

*These statistics are often used by ISPs (Internet Service Providers) to monitor the use of the system. For example, if the **PERCENT** figure for terminal lines (connected to modems) is very high, that shows that the modems are being heavily used. The ISP might need to add more modems to the system. On the other hand, if the **PERCENT** value is low, it could mean that the ISP has more modems than needed, and that hardware resources could be allocated to a higher demand area. The **OFF** variable can be used to indicate if there are a large number of interrupts of some type, which could mean that the terminal devices are having problems. The key point is that the system administrator needs to spend time getting to know what the Daily Reports for a particular server are and then must be able to spot unusually high or low statistics.*

- ④ The second part of the report deals with specific users and the resources that they are utilizing from the system. Before studying these fields, remember that **PRIME** time is set in the text file `/etc/acct/holiday`. Any time other than **PRIME** time is **NPRIME** time.

This section of the report contains the following fields:

UID	The UID of each user. This is the UID found in the second field of the <code>/etc/passwd</code> file. This UID could also come from a NIS, NIS+ or LDAP server that contains UIDs of the network.
LOGIN NAME	The login name of the user. This is also found in the <code>/etc/passwd</code> file or in a NIS, NIS+ or LDAP server.
CPU	Shows the total cumulative minutes that the user ran programs on the CPU. It is broken down into PRIME and NPRIME minutes. Understand that most workstation users don't use a tremendous amount of CPU time. If a user only saves a small file to the hard drive at the end of the day, it is quite possible to see a zero or a very low number. However, demanding programs such as Oracle 9i can have dramatic looking CPU statistics.
KCORE - MINS	Shows the total amount of kilobytes of memory divided by time (minutes). This is the average amount of memory that was used by each user during the that user's connection time. For example, if a user started a program that used 10 MB of physical memory for 5 minutes, quit the program, and logged off 5 minutes later, the average KCORE time would be about 5 MB/Minute. This statistic is also broken down into PRIME and NPRIME .
CONNECT	The total time the user connected to the system during the reporting period, broken down into PRIME and NPRIME .
DISK BLOCKS	The cumulative total disk blocks that the user has used on the system. Remember, a block is 512 bytes or 1/2 KB. To calculate the number of KB-bytes used, divide the blocks by 2. In Figure 28.1, the amounts have been rounded down to zero.
# OF PROCS	The number of processes that the user started. Because this report only shows users how have logged into the system during the accounting period, if this number is zero for a given user, there could be something wrong with this user's account. On the other hand, if the number is extremely high (such as 5000), look into this user's account activity for a runaway program or for user conduct issues.
# OF SESS	The number of sessions (logins) that the user had during the reporting period. . This number should be relatively low, typically less than 20. If a user has a very high number like 200 or 300, that user is probably using a "boot" or script to log in and log out. Check the company's security policy on login scripts. Some companies don't allow this type of automated login. Some software programs, such as email programs, also automatically log in and log out of the system.
# DISK SAMPLES	Number of times the user's file system was checked. This gives the system administrator a feeling for how often the samples were taken. If a user has a database program that performs a lot of reads and writes, the DISK BLOCKS statistic may not give an accurate portrait of the user's true disk usage. Some users might copy a file from their hard drive to the server, perform calculations, and then copy the file back from the server to the hard drive during the sample times.

FEE

The Fee that would be charged to a user if the **chargefee** utility is run. As mentioned earlier, most companies use other software to calculate fees, rather than **chargefee**.

Lesson 28.3 Creating a Daily Report

This lesson will show the reader how to generate a daily report. Unfortunately, this daily report will not have very much information, because it will not much activity to report.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command **vi /etc/holidays**
4. Add this line to the file:
12/25 Christmas
5. Save the file with the command **:wq!**
6. Type the command **/usr/lib/acct/chargefee user11 50**
This command charges user11 an extra 50 units (currency is not defined).
7. Type the command **cd /var/adm/acct/nite**
If lock files exist that could prevent the daily report from being run, the locks are removed by the next three steps.
8. Type the command **rm lock***
9. Type the command **rm lastdate**
10. Type the command **rm statefile**
11. Type the command **date**
For later in this lesson, make a note of the system's day and month in MMDD form. For example, August 30 = 0830 or May 15 = 0515.
12. Login as the adm user with the command **su - adm**
*The **su -** command is used to change one user into another user. In this case, the root user is changed into the adm user. The adm user is the only one that should run the **runacct** command.*
13. Type the command **/usr/lib/acct/runacct**
This turns on System Accounting.
14. Type the command **exit**
*This **exit** command logs off the adm user and returns to the root user.*
15. Log in and out as various users, and run a variety of commands. One good command for generating a lot of disk activity is **find / -name "*" (This says to "find everything.")**.
16. Log in as the root user.
17. Set the system date *two days in the future*, using the **date** command.
This ensures that the Daily Report will have at least one complete 24-hour period to report back on.

The format for this command is:

date -u mmddHHMM

MM = Month

DD = Day

HH = Hour (in 24 hour military time)

mm = Minutes

For example, if the current date is

August 18, 2002

*type the command **date -u 08201230***

This will set the date to August 20, 2002 at 12:30 A.M. Note that the time is required, and is expressed in Greenwich Mean Time; adjust your time entry accordingly.

18. Login as the adm user with the command **su - adm**
*The **su -** command is used to change one user into another user. In this case, the root user is changed into the adm user. The adm user is the only one that should run the **runacct** command.*
19. Type the command **/usr/lib/acct/runacct**
20. Type the command **exit**.
 You should now be the root user.
21. Type the command **id**. If it does not show that you are the root user, log off and then log in again as the root user.
22. Type the command **/usr/lib/acct/prdaily > /mydaily**
23. Type the command **vi /mydaily**
*The report **/mydaily** is the daily report generated by the **runacct** command. Unfortunately, it will only contain data from a few minutes, not from a full 24-hour period. Remember that this lesson was designed to show how to generate a daily report.*
24. When you are done looking at the report, exit from **vi** with the **:q!** command.
25. If desired, use the **date -u** command to reset the proper time and date.

The Daily Command Summary

The Daily Command Summary is the section of the Daily Report that summarizes the resources that each command uses. This is very beneficial for determining what program is using the most server resources. The Daily Command Summary is very useful for spotting users that take up a disproportionate amount of system resources. Also, there are some applications that use too much memory or CPU cycles for a server to handle. Use this report if users complain that the server is "sluggish."

The Daily Command Summary give statistics for each command that has been run by a user during the reporting period. Because it is a system-wide report, it does not indicate which users ran which commands. It reports the "mean average" for many of its statistics. If a program has used 1.50 units of CPU time in 10 runs, the summary reports an average of 1.50/10, or 0.15 units.

For convenience, the Daily Command Summary part of Figure 28.1 is repeated below, followed by a description of each column of this summary. The table below only shows three of the commands; most of this section has been truncated in Figure 28.1.

COMMAND NAME	NUMBER CMDS	TOTAL KCOREMIN	TOTAL CPU-MIN	TOTAL REAL-MIN	MEAN SIZE-K	MEAN CPU-MIN	HOG FACTOR	CHARS TRNSFD	BLOCKS READ
TOTALS	452	13962.20	1.44	41756.81	9682.52	0.00	0.00	1912348032	6979
sun	2	8294.40	0.46	2780.90	17992.19	0.23	0.00	256008192	659
java	1	3570.35	0.14	0.38	24880.46	0.14	0.38	3151360	217

NAME The name of the command.

NUMBER CMDS The number of times this command was run during prime time. For example, if the Netscape browser was started 5 times by a user, this number would be **5**.

TOTAL KCOREMIN The cumulative measurement of memory segments used by the command for each minute of execution. This information is in kilobytes of memory.

TOTAL CPU-MIN The total CPU time in minutes that the command took during prime time. This is CPU time, rather than how much real time while the command was running. Most desktop applications should have little or no CPU minutes. Only powerful applications like Oracle or the Apache Web server should show significant CPU-MIN values.

TOTAL REAL-MIN The total real time during which the command has run. This is the cumulative time for all the times that the command was executed. In the above example, if the Netscape browser

was run 5 times for 8 minutes each, the REAL-MIN column would give the cumulative total of 40.

MEAN SIZE-K	The mean kilobyte segments of memory used for each command. This is the mean of TOTAL KCOREMIN / NUMBER CMDS.
MEAN CPU-MIN	The mean CPU time for each command, derived from TOTAL CPU MIN / NUMBER CMDS or the average CPU time each time the command was run.
HOG FACTOR	The TOTAL CPU-MIN / system up time for this command. This shows how busy the system is. If the sum of the HOG FACTOR for all commands is higher than 90%, it's time to either purchase some more CPUs or schedule jobs so that the existing CPUs are not so busy all the time. The only other alternative is to reduce the number of programs available on the system.
CHARS TRNSFD	Total number of characters transferred by this command. This statistic can be misinterpreted and should be ignored.
BLOCKS READ	Total number of memory blocks the command used. This is a good way to see if a program is a memory hog. The Java Forte Engine is a good example of a memory hog. It tries to capture 1/2 GB of memory for use, even though it does not really need all that memory.

The Monthly Command Summary

The Monthly Command Summary is similar to the Daily Command Summary except that the statistics are gathered for an entire month instead of a day.

Actually, this command gathers statistics from the start of the current fiscal period to the time when it is run. To make it a true Monthly Command Summary, it should be run by the adm user from a **cron** file. As mentioned above, one way to do this is:

```
30 7 1 * * /usr/lib/acct/monacct
```

This command runs **monacct** at 7:30 A.M. on the first day of each month.

The adm user can also run **monacct** from the command line:

```
/user/lib/acct/monacct
```

The **monacct** command produces human-readable output from the raw data files created by **runacct**. By default, these files are located in the directory

```
/var/adm/acct/fiscal
```

and have names such as **fiscrpt09**.

The format of a Monthly Report is identical to the format of a Daily Report, as described above.

The Last Login Report

This report displays the most recent login for each user, by date and userID. It appears in the **LAST LOGIN** section of the Daily Report.

The following table shows the section labeled © in Figure 28.1 (headings have been added). In this listing, there are three Date / User entries on each line:

Date	User	Date	User	Date	User
00-00-00	nobody4	00-00-00	sms-fomd	02-09-05	root
00-00-00	nuucp	00-00-00	sms-frad	02-09-05	user11

This report shows the last login date for each user, in YY-MM-DD format. The users listed with a date of 00-00-00 are system users. The significant entries are in the last two columns. Both the root user and user11 had their last logins on September 5, 2002.

The runacct command

As noted above, the human-readable Daily Report is produced by the command `/usr/lib/acct/prdaily` and the `/usr/lib/acct/monacct` command is used to create the human-readable Monthly Report.

However, the raw system usage data for both of these reports is produced by the `/usr/lib/acct/runacct` command. As mentioned above, this command should be run from the adm user's `cron` file, with a line such as:

```
30 4 * * * /usr/lib/acct/runacct 2> /var/adm/acct/nite/fd2log
```

This file should be run every 24 hours for accurate accounting information. This command can also be used at the command line to show the Daily Report and the Monthly Report.

The `runacct` command uses the following syntax.

```
runacct MMDD [state]
```

MMDD Month and Date to run the report.

[state] The following states can be specified:

CMS Produces the Command Summary Report.

CLEANUP Cleans up temporary files used by the `runacct` command.

CONNECT Displays the Connection Session report.

DISK Merges the disk access records with the process, connect and fee records.

FEES Uses the `acctmerg` program to merge `tacct` records (ASCII format) from the fee file into the `dayacct` file.

MERGETACCT Merges the `daytacct` records with the summary total records in `/var/adm/acct/sum/tacct`

MERGE Merges the process and connect accounting records.

PROCESS Creates the process accounting records. Uses the `tacct.h` format.

SETUP Creates the working files by moving the accounting files into the `/var/adm/night` directory.

USEREXIT	Calls any local accounting program that the system administrator wants to run. This name of this program should be <code>/usr/lib/acct/runacct.local</code> .
WTMPFIX	Checks the condition of the <code>/var/adm/wtmpx</code> file. Fixes this file if there are errors.

Troubleshooting System Accounting Problems

There are times that the System Accounting software will not work properly or will not start. If you have problems:

1. Check the adm user's **crontab** file with the command

```
crontab -e adm
```

There should be three lines in the file with the format

```
0 * * * * /usr/lib/acct/ckpacct  
30 4 * * * /usr/lib/acct/runacct 2> /var/adm/acct/nite/fd2log  
30 7 1 * * /usr/lib/acct/monacct
```

The **cron** file uses the following format:

(1)	(2)	(3)	(4)	(5)	(6)	(7)
(1) – Minute				(Values 0 - 59)		
(2) – Hour in military time				(Values 0 – 23)		
(3) – Day of month				(Values 1 – 31)		
(4) – Month of year				(Values 1 – 12)		
(5) – Specific weekday				(Values 0-Sunday, 1-Monday, 2-Wednesday, 3-Thursday 4-Friday, 5-Saturday and 6-Sunday)		
(6) – Command						(What users type on a command line to run the command)

Use the **date** command to check the system's current date. Understand that the line **30 7 1 * * /usr/lib/acct/monacct** shown above sets up the monthly accounting software to run on the first day of the month at 7:30 A.M. This command should only be run from the adm user's **crontab** file.

2. Check the root user's **crontab** file with the command

```
crontab -e root
```

There should be a line in the file with the format

```
30 22 * * 4 /usr/lib/acct/dodisk
```

Here, the command is set up to run at 10:30 P.M. (military time) on Friday. If the disk is being used heavily during that time, change the time.

3. Check the `/etc/rc0.d/K22acct` and `/etc/rc2.d/S22acct` scripts

Type the command `ls -l /etc/rc0.d/K22acct`

Type the command `ls -l /etc/rc2.d/S22acct`

Both these files should exist and look exactly like the file `/etc/init.d/acct`

4. Type the command

```
/etc/init.d/acct start
```

This command should run the script `/usr/lib/acct/startup`. The `/usr/lib/startup` script should then run the script `/usr/lib/acct/turnacct`.

The `turnacct` script executes the script `/usr/lib/acct/nulladm`. The `nulladm` script sets owner and group to `adm` and then uses the command `chmod 664 mode` to set required file permissions. The file `/var/adm/pacct` should then have the proper permissions

The `turnacct` script also cleans out the files:

```
/var/adm/acct/wtmp*,  
/var/adm/acct/sum/pacct*,  
/var/adm/acct/nite/lock*
```

These files should be deleted immediately before the `/etc/init.d/acct start` command is run.

Check to make sure these files really are deleted. Make sure that the directory `/var/adm/acct` exists and that the files in it don't have write permission problems.

The `turnacct` command also runs the command `acctcon -on $filename`.

The `/var/adm/pacct` file records when the termination of a process occurs. Make sure that the `/var/adm/pacct` file exists. In this case the `acctcon -on $filename` command is using the `/var/adm/pacct` file to process connection time information.

Check to see if all the files referenced in the `/etc/init.d/acct` startup script are present. If critical files are missing:

1. Backup any existing System Accounting files.
2. Reinstall the System Accounting programs:
 - Type the command `pkginfo | grep SUNWaccr`
 - Type the command `pkginfo | grep SUNWaccu`

How to Repair the `tacct` file

The `/var/adm/acct/sum` directory contains the `tacct` file. This file is used to hold the accounting information in binary format.

If this file becomes corrupted:

1. Type the command:

```
cp /var/adm/acct/sum/tacct /tacct.backup
```

This command copies the `tacct` file to a backup file

2. Type the command
`/usr/lib/acct/acctmerg -v /var/adm/acct/sum/tacct<MMDD> / tacct.mytext`
*This command cleans up errors in the **tacct** file.*

3. Type the commands

```
/usr/lib/acct/acctmerg -i < /tacct.mytest > /var/adm/acct/sum/tacct<MMDD>
```

4. If necessary, merge the file **tacctprev** with the **tacct<MMDD>** file with the commands:

```
/usr/lib/acct/acctmerg < /var/adm/acct/sum/tacctprev  
/var/adm/acct/sum/tacctMMDD > tacct
```

The command **prtacct** is used to display the contents of the **tacct<MMDD>** file. The **tacct<MMDD>** file is a binary file that can not be directly edited or changed with the **vi** editor or another text editor.

5. If this repair process seems to make the problem worse, type the command
`cp /tacct.backup /var/adm/acct/sum/tacct`
to restore the file to what it was before.

For further options, connect to <http://sunsolve.sun.com>.

How to Repair the **wtmpx** File

1. Type the command `cp /var/adm/wtmpx /wtmpx.backup`
*This command creates a backup copy of the **/var/adm/wtmpx** file named **/wtmpx.backup**.*
2. To view and repair the **/var/adm/wtmpx** file, type the command
`/usr/lib/acct/fwtmp < /var/adm/wtmpx > wtmpx.mytextfile`
If there are corrupt lines in this file, clean up the file.
3. Open the file **wtmpx.mytext** file in a text editor, such as **vi**.
4. If there are corrupt lines in this file, clean up the file.
5. Type the line
`/usr/lib/acct/fwtmp -ic < / wtmpx.mytextfile > /var/adm/wtmpx`
6. If the **wtmpfix** file still doesn't seem right, type the command `/usr/lib/acct/wtmpfix`.
*This command fixes errors in the **/var/adm/wtmpx** file.*
7. If for some reason the **wtmpx** file becomes severely damaged from the repair, type the command
`cp /wtmpx.backup /var/adm/wtmpx`

Key Points to Remember

System Accounting is a nice feature to use when it comes to “getting a feel for a server.” Most system administrators use System Accounting tools that are more powerful and more accurate than the ones offered by Solaris 9. Remember that the lessons in this chapter give results that only cover small amounts of time; they do not accurately reflect what you would see in an actual Daily Report or Monthly Report. If you use the Solaris 9 System Accounting tools, follow this chapter's recommendations to run them from the **adm** user's **cron** files.

Chapter 29 DNS

Lessons in This Chapter

Lesson 29. 1 DNS Client Setup	29-5
Lesson 29. 2 Setup a Simple DNS Server	29-12
Lesson 29. 3 Troubleshooting a DNS server	29-17
Lesson 29. 4 Using the Apache Web Server with DNS.....	29-19

Introduction

Human beings don't like to use IP addresses for network communications. Because humans do not like to use numbers, hostnames were created. Instead of typing the command `ping 64.124.140.181`, the command `ping sun.com` can be used. The hostnames are matched to IP addresses in the `/etc/hosts` file on a workstation. The only problem is that there are millions of hostnames and IP addresses on the Internet. Even medium-size networks can have hundreds of IP addresses and hostnames. Trying to keep the `/etc/hosts` file updated would be impossible.

All hostnames end with a domain name. A domain name is just like a family name. For example, the last name "smith" could be thought of as a domain name. Each child's name could be thought of as a host name, john.smith, julie.smith, karen.smith. In the same context, hostnames are attached to domain names, such as sun.com: server32.sun.com, dataserer.sun.com or webserver3.sun.com.

A DNS (Domain Name Server) server matches a hostname with an IP address. The server performs the same function as the `/etc/hosts` file, except that it has data for all the hosts on the network. If a host's name or IP address is changed in the DNS server, all the clients on the network will know about the change. Solaris 9 has a built in DNS server. The DNS server is installed is BIND 8.2.4 on Solaris 9.

Note: This chapter is set up to use only one workstation. With only one workstation it is impossible to demonstrate advanced DNS topics, such as DNS zones and master and slave servers.

DNS Terms

Make sure you understand these DNS terms before installing a DNS server.

Zone	A <i>zone</i> controls the DNS name space. For example, at Sun Microsystems the sun.com zone is controlled by Sun. The Intel Corporation has zone control over the intel.com domain. A zone is the central authority over a domain.
Zone Master Server	The server that controls a Zone. All changes for the zone are made in this server.
Zone Slave Server	A server that takes the load off the Zone Master Server. The Zone Slave Server server updates its own zone files but it does not make DNS changes itself. Along with the Zoner Master Server, the Zone Slave Server handles DNS client requests.
Zone Stub Server	This server only holds records needed to locate name servers of a subdomain. This server does not maintain a full database of addresses and hostnames.

The DNS Client

Solaris 9 can be set up to be a DNS client. If DNS is not set up on a Solaris 9 workstation or server, fully qualified domain names like docs.sun.com and sunsolve.sun.com can not be resolved. This means that the Netscape Navigator browser, ftp and other network-based software will not work.

Two files need to be modified so that Solaris 9 can resolve DNS names. These files are `/etc/resolv.conf` and `/etc/nsswitch.conf`.

The `/etc/resolv.conf` file

The file `/etc/resolv.conf` has the format

```
domain <name-of-domain>           (the domain of the server on which it resides)
nameserver <IP-address-of-DNS-server>
nameserver <IP-address-of-2nd-DNS-server> (there can be up to 8 nameserver lines)
search <domain-names-to-search> (alternate domains to search)
```

The `/etc/resolv.conf` file is used to help the system connect to a host. For example, if a user types the command

```
ping server37
```

the system will proceed as follows:

1. It tries to find **server37** (exactly as typed).
2. If it can not figure out the IP address for **server37** it tries the *domain* listed in `/etc/resolv.conf`. For example, if the user is at sun.com, the system looks for server37.sun.com.
3. If it still can not figure out the IP address, it searches for **server37** in each of the domains listed in the `<domain-names-to-search>` field of the line in `/etc/resolv.conf` that begins with **search**. This continues until it finds the IP address or has tried all the names in the **search** line.
4. If it still can not figure out the IP address, it consults the DNS servers listed in the *nameserver* lines (there can be 1 to 8 of these lines).
5. If it still can not find the IP address, it returns the message **unknown host server37**.

The `/etc/nsswitch.conf` file

The `/etc/nsswitch.conf` file tells Solaris 9 how to obtain name service information. In this particular example, the DNS server is used to resolve a host's name to its IP address. It is possible to have a NIS, NIS+ or LDAP server do this. However, most companies use a DNS server for hostname-to-IP-address resolutions because DNS servers specialize in this.

Figure 29.1 shows part of a typical `/etc/nsswitch.conf` file on a Solaris 9 workstation or server. Your system may or may not include the entries shown here.

```
#
# /etc/nsswitch.files:
#
# An example file that could be copied over to /etc/nsswitch.conf; it
# does not use any naming service.
#
# "hosts:" and "services:" in this file are used only if the
```

```
# /etc/netconfig file has a "-" for nametoaddr_libs of "inet" transports.
passwd: files
group: files
hosts: files dns
ipnodes: files
networks: files
protocols: files
```

Figure 29.1 Part of a Typical /etc/nsswitch.conf File

The `/etc/nsswitch.conf` text file tells Solaris 9 where to obtain various kinds of information. Most of the uncommented lines in the `/etc/nsswitch.conf` file match files found in the `/etc` directory.

For example, in an `/etc/nsswitch.conf` file there could be a line.

hosts: files dns

This line tells Solaris 9 to search for a host's IP address in the `/etc/hosts` file. If the host's name is found, an IP address is returned that matches the hostname given. The keyword **files** tells Solaris 9 to look up the information in a local file.

Other possible entries are shown below. Note that the value for the keyword *hosts*: varies widely between systems.

- hosts: nisplus** Use a NIS+ server to resolve hostnames to IP addresses
- hosts: nis** Use a NIS server to resolve hostnames to IP addresses
- hosts: ldap** Use an LDAP server to resolve hostnames to IP addresses
- hosts: dns** Use a DNS server to resolve hostnames to IP addresses
- hosts: files** Use the local `/etc/hosts` file to resolve hostnames to IP addresses

Multiple sources can be specified. If one source does not have the information needed, an alternate source can be used to find the information. For example:

- hosts: nis files** Use a NIS server to resolve hostnames to IP addresses. If the NIS server does not have this information, use the local `/etc/hosts` file to find this information.
- hosts: dns nisplus files** Use the DNS server to resolve hostnames to IP addresses. If the DNS server does not have the information, use a NIS+ server to resolve hostnames to IP addresses. If that does not work, use the local `/etc/hosts` file.

The default action of Solaris 9 is to try one source for information (NIS+, DNS, files) and if that source can not return the information (the source does not exist, or does not know), the next source of information is used. Eventually the answer is found or a **NON-SUCCESS** message is sent to the inquiry.

The behavior of Solaris 9 can be changed. There are four possible states when trying to find information:

- NOTFOUND** The source (DNS, NIS+, NIS, LDAP, local file) responded but did not know the answer.

SUCCESS The source (DNS, NIS+, NIS, LDAP, local file) responded and gave a response.
TRYAGAIN The source (DNS, NIS+, NIS, LDAP, local file) exists but it is busy; try again.
UNAVAIL The source (DNS, NIS+, NIS, LDAP, local file) is not responding or is unavailable.

When one of these states is encountered, there are two possible reactions:

return Stop looking for the information

continue Use the next source

You can specify what action should be taken. For example:

passwd:LDAP [UNAVAIL=return] files If the LDAP server does not respond stop trying to look for the information. In this case, the `/etc/passwd` file would never be consulted for user information.

ethers:NIS+ [TRYAGAIN=return] NIS If the NIS+ server is busy, don't try the NIS server; just quit now.

automount: LDAP [NOTFOUND=return] files If the LDAP server is contacted and does not know the answer, stop. If the LDAP server can not be reached, use local files.

The default behavior of the `nsswitch.conf` file is:

NOTFOUND=continue

SUCCESS=return

TRYAGAIN=continue

UNAVAIL=continue

Do not add these **switch=action** statements to the `nsswitch.conf` file. These **switch=action** statements are the default actions of the resolver.

Comments can be added to the `nsswitch.conf` file by putting a pound (#) symbol in front of the text that forms a comment.

For example, an entire line can be a comment:

```
# This entire line is a comment
```

or, only part of a line can be a comment:

```
hosts: dns files # the last part of the line is a comment
```

When you modify a line in `nsswitch.conf`, it is a good idea to comment out the existing line and create a new line, rather than just deleting the old line. This lets you easily return to the previous state if the modified line causes problems.

For example, before the modification, you might have:

```
hosts: files
```

After the modification, this might be::

```
# hosts: files ← This line is only commented, can be uncommented if the new line does not work properly.
```

```
hosts: dns [UNAVAIL=return] files
```

Understanding `nsswitch` Template Files

There are five `nsswitch` template files in the `/etc/directory`. These files show the proper format for each type of name service:

nsswitch.dns	Example of using DNS for host name resolution
nsswitch.files	Example of using local files for name services
nsswitch.ldap	Example of using an LDAP server for name services
nsswitch.nis	Example of using a NIS server for name services
nsswitch.nisplus	Example of using a NIS+ server for name services

These files are a good reference source, and Sun recommends that you copy them over the `nsswitch.conf` file. However, the author recommends that you *do not* do this.

The problem is that these files are only *generic templates*. They will usually set up all name service requests (such as `hosts`, `ethers` and `netmasks`) to look at only *one* type of name server.

For example, the file `nsswitch.ldap` requires that the LDAP server answer queries from the **networks:**, **protocols:**, **rpc:**, **ethers:**, **netmasks:**, **bootparams:** and **public key:** name services. What if a company uses a DNS server for hostname-to-IP resolution? A system administrator must then create a custom `nsswitch.conf` file for the company's network requirements. These template files are only generic examples that are not unique enough for a particular company's needs.

Lesson 29.1 DNS Client Setup

In this lesson you will set up your system as a client computer for DNS resolution. This lesson requires a connection to an Internet Service Provider.

- If your system is using Microsoft's Internet Connection sharing point, the DNS server is 192.168.0.1 .
- If your workstation is on a LAN, obtain the IP address of your DNS server from a network engineer.

This lesson is based on a Solaris 9 workstation with the following settings:

Hostname	= sun100
IP Address	= 192.168.0.4
Subnet Mask	= 255.255.255.0
Domain	= aassdff.com
Default router	= 192.168.0.1
DNS server	= 192.168.0.1
DNS server	= < Any_Other_DNS_Server_IP_Address >

It's best to use the above settings for this lesson. You can use your actual network specific information, but other chapter lessons will have to be modified accordingly to work properly.

1. Log in as the root user.
2. Open a Terminal window.

To open a terminal window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Tools menu item, then left click on the Terminal window icon.

3. Type the command **cp /etc/resolv.conf /etc/resolv.backup**
This command saves the current copy of the **/etc/resolv.conf** file to a backup file. To restore this file if a mistake is made, type the command
cp /etc/resolv.backup /etc/resolv.conf
4. Type the command **cp /etc/nsswitch.conf /etc/nsswitch.backup**
This command saves the current copy of the **/etc/nsswitch.conf** file to a backup file. To restore this file if a mistake is made, type the command
cp /etc/nsswitch.backup /etc/nsswitch.conf
5. Type the command **vi /etc/resolv.conf**
6. Edit the **/etc/resolv.conf** file so it looks like this:

```
domain      aassddff.com
nameserver  192.168.0.1
nameserver  < Internet Service Provider DNS IP Address
```

The **/etc/resolv.conf** file above is based on the idea that the client has to customize the example. Use the following:

```
domain      domain.com           ← Domain that the workstation belongs to
nameserver  xxx.xxx.xxx.xxx       ← IP address of the test DNS server (this is used with later
nameserver  yyy.yyy.yyy.yyy   ← IP address of the DNS server(s) from your Internet Service
                                Provider or DNS server(s) on the network.
```

7. Save the **/etc/resolv.conf** file
8. Type the command **vi /etc/nsswitch.conf**
9. Edit the file so that the **hosts:** entry so it looks like this

```
# hosts: files ← You need to # comment out the original line.
hosts: files dns ← Create this line exactly as it is shown here.
```

Be sure to comment out the original **hosts: files** line. The new **hosts: files:** line tells the workstation where to look for hostname-to-IP-address resolution. The first keyword **files** tells the workstation to look in its **/etc/hosts** file first when resolving an unknown host name. The second keyword **dns** tells the workstation to try and find an IP address by contacting the DNS server(s) listed in the **/etc/resolv.conf** file.

10. Save the **/etc/nsswitch.conf** file.
11. Type the command **ping -s uswest.com** (use a different host, if you want)
The screen should show output like:
PING uswest.com: 56 data bytes
64 bytes from qwest.com (199.168.32.62): icmp_seq=0. time=316. ms
64 bytes from qwest.com (199.168.32.62): icmp_seq=0. time=316. ms
64 bytes from qwest.com (199.168.32.62): icmp_seq=0. time=316. ms
12. Press CTL + C simultaneously to stop the ping.

If the ping worked, it means that dns address resolution is now working on your workstation. The Netscape browser that comes with Solaris should now be able to display webpages.

Files Used with a DNS Server

There are five files that are necessary for a DNS server to work properly. These files are not too long, so they can be typed by hand. You can also copy them from this book's official Web page, <http://www.teachmesun.com>.

The five critical file are covered below.

The `/etc/named.conf` File

`/etc/named.conf`

This is the primary configuration file used with a DNS server. This file points to the location of the other four configuration files. This file is the jump-off point for the `in.named` daemon (this daemon *is* the DNS server).

Figure 29.2 is an example of a typical `/etc/named.conf` file. The circled numbers have been added at labels. They do not appear in the actual file.

```
① options {
    directory "/var/named";
};

② zone "." in {
    type hint;
    file "named.root";
};

③ zone "aassdfff.com" in {
    type master;
    file "domain-info";
};

④ zone "0.168.192.in-addr.arpa" in {
    type master;
    file "inverse-domain-info";
};

⑤ zone "0.0.127.in-addr.arpa" in {
    type master;
    file "loopback-domain-info";
};
```

Figure 29.2 A Typical `/etc/named.conf` File

Let's examine the `/etc/named.conf` file in detail.

Note: All commands and organization elements (except `#` comments) end in the semi-colon (`;`) character.

① The first three lines are:

```
options {
    directory "/var/named";
```

```
};
```

The `/var/named` directory is the location for all DNS files. This directory does not exist by default. You must create this directory for the DNS server to work. However, you can use any directory, and you can change that directory, as long as the entry shown above is changed to match it. For example, `/usr/user11/mydnsfiles` is a valid entry.

② The next four lines of the `/etc/named.conf` file deal with the Internet's root domain:

```
zone "." in {  
    type hint;  
    file "named.root";  
};
```

The first line in this defines the *zone* called the *root domain* is the top level domain of the Internet. It is maintained by Internic. This domain defines the subdomains, such as `.com`, `.net` and `.gov`.

Understand that a DNS server reads an Internet address from *right to left*. In a Web address, the placeholder for the root domain is a period (`.`) character. For example, `www.cnn.com`. has a period after end of the `com` domain. Most browsers don't require a user to type the last period, but technically that is the root domain. Try including the final period on a Web address! Another example of a final period DNS name is `http://www.sunfreeware.com.` (with a period at the end). This is a valid DNS entry.

The line `type hint;` tells Solaris 9 that if a domain is not known by this DNS server, it should contact the Internet *A root servers*, also known just as *root servers*. The line `file "named.root";` tells the DNS server to read `/var/named/named.root` as a reference for the *A root servers*. This file contains the IP addresses and settings of the *A root servers*. It must be downloaded from Internic before a DNS server will run properly. The address is `ftp://ftp.rs.internic.net`.

So, this section basically does the following:

```
zone "." in { ← The dot refers to the root domain.  
type hint; ← This indicates that a hint zone is being used to specify the root domain. If the domain is not  
              known, the DNS service must refer to the A root servers.  
file "named.root" ← For information on the A root servers, look in the named.root file.
```

③ The next four lines deal directly with the local domain. In this case, the local domain is `aassddff.com`:

```
zone "aassddff.com" in {  
    type master;  
    file "domain-info";  
};
```

This section tells the DNS server that the master file for the `aassddff.com` domain is a file with the name `domain-info`. This file defines the master and slave DNS servers, and the hostnames and IP addresses of DNS clients in the domain. You can use any filename, and can change that filename, as long as the entry shown above is changed to match it. This means that the filename `domain-info-mydog` is correct if there is a file in the DNS directory `/var/named` with the name `domain-info-mydog` that contains the proper DNS syntax.

```
zone "aassddff.com" in { ← This section indicates that this zone is the aassddff.com domain.  
type master ← This indicates that the server for this zone is authoritative and maintains the master copy  
              of the data for the zone.  
file "domain-info" ← This file holds many of the settings for the aassddff.com domain.
```

④ The next four lines in the `/etc/named.conf` file are:

```
zone "0.168.192.in-addr.arpa" in {
    type master;
    file "inverse-domain-info";
};
```

This section is used when an IP address is known, but the hostname that goes with it is not. This is known as a *reverse lookup*. Here, the numbers **0.168.192** are actually the network number **192.168.0** placed in reverse order. Because this is a reverse lookup, all IP addresses must be put in reverse order. The filename "**inverse-domain-info**" is a text file in the `/var/named` directory. This filename can be changed, as long as the entry for it above is changed, so the name could be changed to "**mydog-reverse-domain-info**" if desired.

```
zone "0.168.192.in-addr.arpa" in { ← This section indicates that the zone to use for reverse lookup is
0.168.192.
type master; ← This indicates that the server for this zone is authoritative and maintains the master copy
of the reverse lookup data for the zone
file "inverse-domain-info"; ← This file holds the reverse lookup table for this zone.
};
```

⑤ The next four lines are :

```
zone "0.0.127.in-addr.arpa" in {
    type master;
    file "loopback-domain-info";
};
```

This section is used for loopback tests, like `ping 127.0.0.1`. The file **loopback-domain-info** can be renamed to any filename. The file must reside in the dns directory `/var/named`. There are other types of information that can be placed in a **named.conf** file.

```
zone "0.0.127.in-addr.arpa" in { ← This section indicates that the zone to use for loopback tests is
0.0.127.
type master; ← This indicates that the server for this zone is authoritative for loopback tests.
file "inverse-domain-info"; ← This file holds the loopback information for this zone.
};
```

Before we go on, a simple way to understand the DNS structure is:

- The `/etc/named.conf` file points to the configuration directory and DNS configuration files.
- The DNS configuration files are in the `/var/named` directory.

DNS Configuration Files

Now let's look at the DNS Configuration files. These are defined by the `/etc/named.conf` file.

`/var/named/named.root`

This file must be downloaded from the `ftp://ftp.rs.internic.net`. It points to the official Internet A servers. These servers are run by Internic and point to the various domains, such as `.com` and `.net`. This file must be present for DNS to recognize Internet domains. For example, the domain `sun.com` is registered on an A server. When a user types the command `ping sun.com`, the IP address for `sun.com` comes from an Internic root server.

/var/named/domain-info

This file defines the SOA (Start of Authority) for this domain. The SOA maps the hostnames to IP addresses in this domain. For example, if there was a server named **data32** on the domain johnson.com, this file would have the following lines:

```
@      IN SOA dnserver1.johnson.com dnserver1.johnson.com (  
  
      IN NS dnserver1.johnson.com ; primary DNS server  
  
      data32 132.16.35.103 ; IP address of the data32 server
```

/var/named/inverse-domain-info

This file is used when an address is looked up to find a matching host. This is the exact opposite function of the domain-info file. Entries in this file reference the last set of digits in an IP address, such as the 103 in 132.16.35.103.

```
4      IN PRT data32.johnson.com
```

/var/named/loopback-domain-info

This file has the same basic format for every DNS server. It has two critical lines.

```
      IN NS  yourhostname.yourdomainname.com  
1     IN PRT localhost.domainname.com
```

Types of DNS Resource Records

The standard DNS Resource records are used in all DNS files. These records are:

- A** Address. This is an Internet Address that defines name-to-IP-address resolution. For example, **mycomputer.mydomain.com 132.24.13.103** would be an A record.
- CNAME** Canonical name, also known as a nickname or alias name. This is a pointer to a real name. For example, the following line would show that **myfax.sun.com** is an alias for **host9.sun.com**:
myfax.sun.com in CNAME host9.sun.com
- HINFO** Host Information. Information about a host.
- MX** Mail Exchange. This gives the IP address of the e-mail server. This should point to an A record.
- NS** Name Server. This record specifies what server on the network is providing DNS resolution.
- PTR** Pointer. This is the key DNS record that resolves an IP address to a host name. Pointers are only used in reverse domain lookups or reverse lookups.
- SOA** Start of Authority. This defines the top of the domain. A name database must start with an SOA. This identifies the best source of information for data in a domain. This has the current version of the DNS database.
- TXT** Text. Associates a text comment with a host name. For example, **sun23.aassdfff.com "The FTP server for the company"** associates the quoted text comment with the address that precedes it.
- WKS** Well Known Service. This is an experimental keyword that points an Internet service to an IP address. Just like MX points to a mail server, HTTP can be pointed to a proxy server.

The following section gives more detail about some of the DNS Resource records listed above.

A The address of a given system (can be a workstation or server). This has the syntax
machine-name TTL class A IP-address

machine-name Hostname of the system
TTL Optional Time To Live in seconds
class Class of the entry, IN = Internet Name is the most common
A The A reserved keyword
IP-address IP address of the system

CNAME

The canonical name of the system. Basically, this can be thought of as a nickname for the system's host name. Technically, this record is a pointer from a name to an alias name. This resource record has the following format:

nickname TTL class CNAME canonical-name

nickname The nickname of the system
TTL Optional Time To Live in seconds
class Class of the entry, IN = Internet Name is the most common.
CNAME The CNAME reserved record.
canonical-name "Real name" such as **server83.sun.com**

HINFO

Information on the host. This can provide information like the host's operating system (Linux, Solaris and so forth) Because this provides information to hackers, many companies do not use this field.

optional-name TTL class HINFO hardware OS

optional-name A "nickname" for the system
TTL Optional Time To Live in seconds
class Class of the entry, IN = Internet Name is the most common
HINFO The HINFO reserved record
hardware Type of hardware, such as **SPARC, Intel32, anyname_desired**
OS Operating system, such as **SUN, Win2000, name_made_up**

NS

This resource record defines the server responsible for a domain. It has the syntax:

domain-name TTL class NS name-server

domain-name The domain name being defined; for example, sun.com or uswest.com.
TTL Time To Live in seconds
class Class of the entry, IN = Internet Name is the most common.
NS The NS keyword
name-server Name of the server responsible for the domain, in the format
hostname.domain.name

SOA This resource record is used to define a zone. It has the syntax:

zone-name class SOA origin email-zone-admin (
serial#
refresh

retry
expire
ttl
)

zone-name Name of the zone (domain) . The zone name **must** end with a trailing period. For example, **aassddff.com.** would be a valid entry.

class Class of the entry, IN = Internet Name is the most common.

SOA SOA (Start of Authority) keyword.

origin The fully qualified domain name of the host that keeps these records. For example, **dns32.sun.com.** is a valid origin name. Notice the domain name **sun.com.** ends with a period!

email-zone-admin The email address of the zone master. Instead of the usual email format johnsmith@aol.com , this has the format **johnsmith.aol.com** for the address. If something goes wrong with the DNS server, this is the contact email address.

serial# Used by slave servers to check if the record is current. If a DNS file is updated, remember to increase this number in the text file. This lets the DNS slave servers know that a record has been updated.

refresh Time in seconds that a slave DNS server should check the Master DNS server for updates.

retry Time in seconds that a DNS slave server should try the DNS master server if the last contact was bad.

expire Time in seconds that the information is considered valid before a DNS slave server should update its records.

ttl Time to Live. This record is used as a default time to live in case a resource does not have a Time to Live explicitly defined in its resource record.

WKS Well Known Service. This is an experimental keyword that is designed to associate an Internet service (such as FTP or HTTP) with a given IP address. This supports the following format:

optional-name TTL class WKS address protocol-list

optional-name name_made_up

TTL Time To Live in seconds

class Class variable, IN= Internet Name

WKS The WKS keyword

address IP address of the service

protocol-list List of protocols that are used with this service, such as RPC, FTP, SNMP

Lesson 29.2 Set Up a Simple DNS Server

WARNING

This lesson sets up a simple DNS server, which is also used in Lesson 29.3 and in Lesson 29.4. Setting up this simple server may change your system's ability to resolve hostnames outside the workstation.

Before you follow this lesson, back up your DNS files:

Type the following commands:

```
cp /etc/nsswitch.conf /etc/nsswitch.backup
cp /etc/resolv.conf /etc/resolv.backup
cp /etc/hosts /etc/hosts.backup
cp /etc/netmask /etc/netmask.backup
cp /etc/defaultrouter /etc/defaultrouter.backup
cp /var/named/<any_dns_file> /var/named/<any_dns_file.backup>
```

When you are ready to restore your previous DNS setup, follow the procedure in Lesson 29.5, "Restoring Your Previous DNS Server Settings."

This lesson gives all the steps necessary to set up a simple DNS server. These files can be typed in by hand, or they can be copied from this book's official webpage, <http://www.teachmesun.com>.

These files must be modified for your system's hostname, domain and IP address. They will work with a system that uses a class "C" subnet mask. The DNS server should respond to the command

```
ping <hostname.yourdomainname.com>
```

If this does not work, see Lesson 29.3, "Troubleshooting a DNS Server."

Note: This lesson assumes that you have set up your system as follows:

Hostname:	sun100
IP Address:	192.168.0.4
Subnetmask:	255.255.255.0
Default Router:	192.168.0.1
Domain:	aassddff.com

You must perform this setup before starting this lesson, *or* substitute your system's actual settings below, as needed.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command `ping sun100.aassddff.com`
4. Proceed as follows:

- *If you see the message:*
sun100.aassddff.com is alive
it means that the system's FQDN (Fully Qualified Domain Name) is being resolved by a DNS server.

If you see this message, you do not need to continue with this lesson. You may skip to the next lesson, or you may continue with this one..

- *If you see the message*

ping: unknown host sun100.aassdfff.com

you must create a simple DNS server with the following steps:

*Instead of typing in each of the required DNS files, follow these steps to copy the file named **files.tar** from this book's official webpage <http://www.teachmesun.com>.*

5. Log in as the root user.
6. Open a Terminal window.
7. Type the command **cd /var**
8. Type the command **mkdir named**
9. Type the command **chmod 777 /var/named**
*The **/var/named** directory is used by the DNS server. The previous steps created this directory with full read/write/execute permissions for everyone. On a production level server, only the root user would have read/write/execute permission to the **/var/named** directory. This lesson sets full permissions for everyone, to avoid any directory permission problems.*

*The <http://www.teachmesun.com> webpage has a file named **files.tar**. Download this file into the **/var/named** directory.*

*If you download **files.tar** to a Microsoft Windows based computer, open an MS-DOS command prompt and then **ftp** the file to the Solaris workstation.*

10. Once the **files.tar** file is in the **/var/named** directory, type the command **tar -xvf files.tar**
*The **files.tar** file has some extra files from Chapter 2. This should not harm the DNS server. For the remainder of the lesson, either copy the files extracted from **files.tar** to the appropriate directories, or edit the system's current files to include or add lines from the extracted files to the matching system files. Understand that this will change your workstation's network settings.*
11. Copy or modify the **/etc/hosts** file so that it has these lines:
127.0.0.1 localhost
192.168.0.4 sun100 loghost
12. Copy or modify the **/etc/netmasks** file so it looks like this:
192.168.0.0 255.255.255.0
13. Copy or modify the **/etc/defaultrouter** file so it looks like this:
192.168.0.1
14. Type the command **hostname sun100**
*The **hostname** command sets the system's hostname in several **/etc** files.*
15. Type the command **echo "aassdfff.com" > /etc/defaultdomain**
16. Type the command **domainname aassdfff.com**
17. Modify the **/etc/nsswitch.conf** file so that the line
hosts: files
is changed to
hosts: files dns
18. Create or modify the **/etc/resolv.conf** file so it looks like this:
domainname aassdfff.com
nameserver 192.168.0.4
19. Create or modify the **/etc/named.conf** file so that it looks like this:

```

options {
directory "/var/named";
};
zone "." in {
    type hint;
    file "named.root";
};
zone "aassdfff.com" in {
    type master;
    file "domain-info";
};
zone "0.168.192.in-addr.arpa" in {
    type master;
    file "inverse-domain-info";
};
zone "0.0.127.in-addr.arpa" in {
    type master;
    file "loopback-domain-info";
};

```

20. If it does not exist, create the `/var/named` directory.
21. Create or copy the `/var/named/domain-info` file so that it looks like this :

```

@ IN SOA sun100.aassdfff.com. hostmaster.sun100.aassdfff.com. (
    1      ; Serial number
    43200 ; Refresh timer - 12 hours
    3600  ; Refresh timer - 1 hour
    604800 ; Expire timer - 1 week
    86400 ; Expire timer - 1 day
)

; Define the name servers for the domain
    IN NS sun100.aassdfff.com ; primary
; Name to IP address mappings for this domain.
    sun100 IN A 192.168.0.4
; Loopback domain definition
localhost IN A 127.0.0.

```

22. Create or copy the `/var/named/inverse-domain-info` file so that it looks like this:

```

@ IN SOA sun100.aassdfff.com. hostmaster.sun100.aassdfff.com. (
    1      ; serial number
    43200 ; refresh timer - 12 hours
    3600  ; refresh timer - 1 hour
    604800 ; expire timer - 1 week
    86400 ; minimum timer - 1 day
)

; Define the name servers

    IN NS sun100.aassdfff.com. ; primary name server

; Define addresses to name mappings in this domain
; last digits in the IP address

    4      IN PTR sun100.aassdfff.com.

```

23. Create or copy the `/var/named/loopback-domain-info` file so it looks like this:

```
@ IN SOA sun100.aassddff.com. hostmaster.sun100.aassddff.com. (  
    1      ; serial number  
    43200 ; refresh timer - 12 hours  
    3600  ; refresh timer - 1 hour  
    604800 ; expire timer - 1 week  
    86400 ; minimum timer - 1 day  
    )  
; Define the name servers  
    IN NS  sun100.aassddff.com.; primary name server  
; Define addresses to name mappings in this domain  
; last digits in the IP address  
1      IN PTR sun100.aassddff.com. @ IN SOA sun100.aassddff.com.
```

If you have not done so, do so now, or you can also download it from the internic.net webpage.

24. Create or modify the `/var/named/named.root` file.

- *If you downloaded the `files.tar` file from this book's official webpage, you should already have extracted it in step 10 of this lesson. If so, skip to step 25.*

Note: This version has been modified for this lesson, but it may not have the latest DNS server information from Internic.

- *You can also download this file from the internic.net webpage. This version will have the latest DNS server information from Internic, but you will need to modify it for use with this lesson.*

To download the latest `named.root` file from Internic, follow these steps:

- a. Use your Web browser to connect to
`ftp://ftp.rs.internic.net/domain/named.root`
- b. In your Web browser, highlight and copy all the text.

CAUTION: In the following steps, be sure to use a *plain ASCII* text editor. *Do not* use a word processing program such as *Microsoft Word* or *WordPerfect*. Word processing programs add invisible special characters to files. This can make the saved file unreadable your Sun system.

- c. Using a *plain ASCII* text editor, such as **vi** or Microsoft's **Notepad**, copy this text into a blank file.
- d. Modify the **A.ROOT-SERVER.NET.** line in the `/var/named/named.root` file so it reads:
`A.ROOT-SERVERS.NET. 3600000 A 192.168.0.4`
This tells the DNS server that this workstation is an A root server. Obviously, this is false, but for this example it is necessary. The book is designed to work even with workstations that are not connected to the Internet. If it is not connected to the Internet, the workstation needs to pretend that it is an A root server so that the DNS server will work.
- e. Save this file on your Sun system as `/var/named/named.root`
Make sure that your editor does not add a file extension such as `.txt` to this file!

25. Type the command `init 6`

The command `init 6` will reboot the server.

26. Type the command `/usr/sbin/in.named`

The command `/usr/sbin/in.named` starts the `in.named` process. This process is the dns server!

27. Type the command `ping sun100.aassddff.com`

- If the command `ping sun100.aassddff.com` responds with the message:
sun100.aassddff.com is alive
The dns server is working properly and you can continue with the rest of the chapter.
- If the DNS server is not working properly go to the next lesson, *Troubleshooting DNS*.

Lesson 29.3 Troubleshooting a DNS Server

This lesson introduces the reader to some common DNS troubleshooting techniques. Understand that DNS is very sensitive to even the smallest typing mistakes.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command `pgrep in.named`
*This command should report back the PID (Process IDentification) number of the **in.named** daemon. If the **in.named** daemon is not running, the DNS server is not running. If the **in.named** process is running but the server can not resolve a name like **mycomputer.mydomain.com** there is something wrong with the dns server's configuration files.*
4. Type the command `pkill in.named`
*This command kills the **in.named** process. Even if the **in.named** daemon is running, its configuration information may be defective so let's kill the process now. Remember, the **in.named** process is the DNS server.*
5. Type the command `ps -ef | grep in.named`
*This command will show if any additional **in.named** processes are alive. If nothing shows up after the command is typed, all the **in.named** daemons are dead. Understand that several **in.named** processes can be running at the same time on a faulty DNS server.*
6. Repeat steps 4 and 5 until no more **in.named** processes are running.
7. Check all the DNS files very carefully. As necessary, modify and save these files.
 - a. Check for typing errors and for extra spaces at the ends of lines.
 - b. Make sure that there is a semi-colon (;) at the end of all lines that are supposed to have one.
 - c. If text was saved from Microsoft Word or some other advanced word processor, it may contain non-ASCII characters that the Solaris system cannot handle. If this has happened, the only real solutions are to recreate the damaged files by hand, or to copy known good DNS files and modify them to match your network settings.
8. Type the command `ping <dns_server's_IP_address>`
for example:
ping 192.168.0.4
This is the addresss that has been set up for this lesson. Also, try pinging any other DNS addresses that you know.
The client must be able to contact the DNS server's IP address through the network. If the client can not ping the DNS server, there is something wrong with your network or Solaris 9. The next couple of steps attempt to troubleshoot network problems.
9. Type the command `cat /etc/hosts`
The /etc/hosts file should look something like this:
127.0.0.1 localhost
192.168.0.4 sun100 loghost
Try adding the hostname and IP address of the DNS server to this file. Also, try adding the DNS client's own hostname and IP address to this file.
10. Type the command `cat /etc/netmasks`
You should see something like:
192.168.0.0 255.255.255.0
Check this network setting with other /etc/netmasks files on servers in the same network area.
11. Type the command `cat /etc/defaultrouter`

You should see something like:

```
192.168.0.1
```

This shows the IP address of the router.

12. Now type the command **ping <router_IP_address>**

This should be the router that the test workstation's Ethernet card is directly attached to. If there is a switch or other network device between the router and the Ethernet card, check the settings and cabling between the server and the router.

13. Type the command **cat /etc/resolv.conf**

You should see something like:

```
domain aassddff.com
nameserver 192.168.0.4
```

Check the domain and nameserver addresses very carefully. The nameserver variable should have the IP address of the dns server(s). Delete any lines that reference a dead or removed DNS server.

14. Type the command **vi /etc/nsswitch.conf**

Just modify the **hosts:** entry so it looks like:

```
hosts: files dns
```

Don't modify the rest of the file.

15. Type the command **init 6**

The command **init 6** reboots the server in a graceful manner.

16. Type the command **pgrep in.named**

This command shows if the **in.named** process has started. This process will start if the **/etc/named.conf** file exists.

17. If the process is not running, make sure the **/etc/named.conf** file exists, and set its permissions with the command

```
chmod 777 /etc/named.conf
```

18. Type the command **rm /var/named/named.run**

This command will destroy the last **named.run** file. Understand that the **/var/named/named.run** file is a debugging file that will be recreated in the next step.

19. Type the command **/usr/sbin/in.named -d 3**

This command will start the DNS server in debug level 3.

20. Type the command **vi /var/named/named.run**

The following display shows some of the more important lines, as they appear on the author's SunBlade 100 server. Unfortunately, the **/var/named/named.run** file can be over 500 lines long, so only selected lines are shown below. You might have to dig through the file to find them. Lines shown in italics below are part of this lesson; they do not appear in the file itself.

Debug level 3

```
Version = in.named BIND 8.2.4 Sat Apr 6 14:44:58 PST 2002
```

```
Generic-5.9-May 2002
```

Make sure that this is the proper BIND version. Sometimes senior level system administrators will reinstall a different version of BIND on a server.

```
confdir = /etc/named.conf
```

Is the configuration file **/etc/named.conf**? Was the file recognized?

```
source = named.root
```

```
reloading hint zone
```

Does these lines appear in the file anywhere?

```
source = domain-info
```

```
source = loopback-domain-info
```

```
source = inverse-domain-info
```

```
source = named.root
```

These lines should appear somewhere in the file for each of the DNS configuration files.

db_load could not open: domain-info: No such file or directory

*If this line appears anywhere in the file, that indicates that for some reason, the **in.named** daemon could not find or open this file. If you see this message:*

- *Check the **/etc/named.conf** file for the exact name of the file.*
- *Make absolutely certain that the **/var/named** directory exists and see if the missing file exists.*
- *If the file is damaged or missing, recreate it with a text editor, or copy it from a known good file.*

master zone "0.0.127.in-addr.arpa" (IN) rejected due to errors (serial 1)

You may see a line that includes the words "error" or "errors" (like this one). Unfortunately, there could be hundreds of different errors, so it's not possible to discuss each one. One good source for information on DNS errors is <http://sunsolve.sun.com>.

```
d='B.ROOT-SERVERS.NET', c=1, t=1, ttl=3600000, data='128.9.0.107'
```

*This line tells you what a root server the **in.named** daemon is trying to contact. Try to ping the IP address listed with some of these servers. If a server does not respond, you have a network problem or a problem with your TCP/IP stack inside the server.*

Using the Apache Web Server with a DNS server

The Apache Web server is included with Solaris 9. This software displays webpages when a request comes from a browser. Any company that displays webpages must have a fully qualified domain name. This section briefly describes the Apache Web server. This server will be used to illustrate various DNS settings.

Lesson 29.4 Using the Apache Web Server with DNS

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command **cd /etc/apache**
4. Type the command
cp /etc/apache/httpd.conf-example /etc/apache/httpd.conf
*This creates the file **httpd.conf** if it doesn't exist.*
5. Type the command **vi /etc/apache/httpd.conf**
*If you make a mistake when editing **httpd.conf**, repeat step 4 and continue from there.*
6. In the **vi** editor, type the command **/Servername**
*When a word is preceded with the frontslash (/) the **vi** editor searches through the text for the location of that word. The command **/Servername** puts the cursor on the first instance of the word **Servername**.*
7. Set this variable to the IP address of the Solaris 9 test server: **Servername 192.168.0.4**
8. Save the file and exit the **vi** editor.
9. Type the command **/usr/apache/bin/httpd**
10. Start the Netscape browser.
11. Enter the URL: **http://sun.100**
or
Enter the URL: **http://your_host_name**
You should now see a webpage that begins with the message:
If you can see this, it means that the installation of the Apache web server software on this system was successful. You may now add content to this directory and replace this page.
12. Enter the URL **http://sun100.aassddff.com**
or
http://your_host_name.your_domain_name

If you can call up one of these webpages, the the DNS server is running.

Lesson 29.5 Removing DNS from a Server

This lesson shows the reader how to remove a DNS server from a Solaris system.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command **pkill in.named** several times.
*The command **pkill in.named** is used to kill the **in.named** process. Remember that the **in.named** process is the DNS server.*
4. Type the command **cp /etc/named.conf /etc/named.backup**
*This command saves the **named.conf** file as a backup file just in case you want to restore the DNS server in the future.*
5. Type the command **rm /etc/named.conf**
*This command removes the **/etc/named.conf** file from the system. When the system is rebooted, the **in.named** process will not start if the **/etc/named.conf** file is not present.*
6. Proceed as follows:
 - *If you followed the warning advice in Lesson 29.2 to back up your networking files:*
 - a) Type the command **cp /etc/hosts.backup /etc/hosts**
 - b) Type the command **cp /etc/nsswitch.backup /etc/nsswitch.conf**
 - c) Type the command **cp /etc/netmask.backup /etc/netmask**
 - d) Type the command **cp /etc/defaultrouter.backup /etc/defaultrouter**
 - e) Type the command **cp /etc/resolv.backup /etc/resolv.conf**
 - *If you did not follow the warning advice in Lesson 29.2 to back up your networking files:*
 - a) Edit the **/etc/resolv.conf** file and make sure the IP addresses in the command point back to your previous DNS server's IP address.
 - b) If you want to use only local files are for hostname to IP address name resolution, type the command **cp /etc/nsswitch.files /etc/nsswitch.conf**
7. Type the command **cd /var**
*Warning: The following command will completely remove the **/var/named** directory. Save these files to an alternate location (if desired) before typing this command.*
8. Type the command **rm -rf named**
9. Type the command **ping sun100.aassdfff.com**
*If the DNS server has been properly removed from the system, the **ping** command should produce an error message:*
sun100.aassdfff.com host not found
This indicates that the DNS server is not resolving the hostname sun100.aassdfff.com.

Key Points to Remember

This chapter demonstrated the basics of DNS. Most system administrators find a good Solaris DNS server on the network and copy all the configuration files to a new DNS server. After the files are copied to the new server, the DNS files are modified to meet the new DNS requirements.

Unfortunately this chapter is set up to use only one workstation. With only one workstation it is impossible to demonstrate advanced DNS topics, such as DNS zones and master and slave servers.

Chapter 30 LDAP

Lessons in This Chapter

Lesson 30.1 Configuring the Directory Server Included with Solaris 9	30-5
Lesson 30.2 Working with the iPlanet Console	30-18
Lesson 30.3 Creating a Console Startup Script and Changing the Console View	30-22
Lesson 30.4 Creating a Configuration Administrator with Console	30-24
Lesson 30.5 Administration Server Basics	30-30
Lesson 30.6 LDAP Save, Restore, Export and Import	30-41
Lesson 30.7 More LDAP Administration	30-43
Lesson 30.8 Create Another LDAP Administrator	30-46
Lesson 30.9 Monitoring the Directory Server's Performance	30-49
Lesson 30.10 Using Command Line Utilities	30-51
Lesson 30.11 Using the Netscape Browser with LDAP	30-52
Lesson 30.12 Set Up Alternate Run Control Scripts (optional)	30-54
Lesson 30.13 Set Up a Client Profile	30-58
Lesson 30.14 Uninstall the Directory Server	30-62

Introduction

LDAP (Lightweight Directory Access Protocol) is an industry standard protocol used to access directory services. Solaris 9 has a suite of directory tools and programs that is called the Sun ONE Directory Server. The current release for Solaris 9 is version 5.1. This features a GUI interface and a number of useful command-line utilities that support the LDAP v3 protocol. Along with the LDAP protocol, Solaris 9 uses a backend database that can be configured to use LDAP to store and retrieve information about users, services and hosts.

LDAP servers can store almost any company-wide information, such as phone numbers, email directories, team schedules, group calendars, network information, hostnames, passwords and telephone numbers. The best data to store in an LDAP directory is data that is relatively static, such as employee names and office locations. Under this use pattern, most LDAP servers are optimized for read access. Updates are infrequent because they use a lot of system resources. This is where a database program like Oracle 9i and a directory service differ in operations. However, just like any databases, LDAP servers range from huge Internet servers like InfoSpace and Bigfoot, to company-wide LDAP servers that are a mix of internal and public information, to private and highly secure LDAP servers.

It's important to note that LDAP is a protocol, not a database. Like SQL (Structured Query Language), LDAP is independent of the data storage mechanisms underneath it. In theory, any database could be fitted with an LDAP server interface so that a client could "speak" LDAP to the server. The LDAP server would then send the data to the underlying operating system and database program, such as IBM DB2, BerkeleyDB, Oracle, or even a series of flat text files. With LDAP, this is all done in a stable and commonly understood way, so that different systems can talk to each other. Unlike SQL, however, LDAP is a full Internet protocol its own right that rides over TCP and has specific methods of encoding special characters and controlling connections.

That being said, it's also important to note that, unlike most databases, LDAP servers generally come with a predefined hierarchy of data types and groups. If you overhear a discussion peppered with references to terms like "organizational unit," "ou=People" and "inetOrgPerson," they're probably talking about LDAP. One reason that LDAP has become popular is that it is not just the *protocol* that is standardized, the *data types* are also extendable

for a company's own purposes. (These data types came from the X500 standard, which is described below.) This is invaluable when trying to unify heterogeneous environments of different types of clients and servers.

Why Switch to LDAP

LDAP servers are being used to store naming information because they can replace the small internal databases that some client utilities try to create on their own. An email client is a prime example of this, because most email users create their own email directories. With an LDAP server, the email directory is set up on a corporate basis and everyone's email client is consistent. Any application that needs to authenticate users, store user-specific information or locate hosts on a network by name can benefit from centralized directory services. This also gives a company central administration of its information, making it easier to update and maintain. Confidential information (such as Social Security numbers, home telephone numbers and garnished wages) can also be tightly monitored and restricted.

LDAP's goals are similar to the original goals of NIS (Network Information Services) and NIS+ . These services were designed to replace static text files stored on individual workstations and servers with something centralized that can scale to a company's structure as it grows. It cannot be said that they were entirely unsuccessful. However, for a number of reasons, Sun Microsystems is moving away from NIS and NIS+ for the storage of user and host information. One reason is that LDAP is an Internet standard and as such is more interoperable among differing operating systems and platforms. Also, *replication* (automatic distribution of naming information between an LDAP server's primary store of naming information – the *master*—and backup or distributed copies—the *replicas*) is much faster than the equivalent distribution process under NIS or NIS+.

Sun LDAP Features

The LDAP server in Solaris 9 is a hierarchical namespace server. Information is stored in a tree-like structure, rather than in a flat or unstructured group of files or tables. Solaris 9 supports both LDAP *Master Servers* and *Replica Servers*. For large corporations, the LDAP server supports a multi-master LDAP topology, in which more than one Master Server can exist. In traditional LDAP replication, a single Master Server can handle read and write requests to the directory, while replicas can only work with read requests.

Because LDAP servers can be used to provide authentication services, security is extremely important. The LDAP server in Solaris 9 supports SSL (Secure Sockets Layer). In addition to SSL security, the Solaris 9 LDAP server also uses TLS (Transport Layer Security). Clients can connect under anonymous, proxy anonymous and proxy connections. Access to data can be restricted, based on the time of day, by the client's IP address, or even by the client's DNS name.

LDAP clients can access the LDAP server anonymously, or they can bind to the directory as a pre-existing LDAP object. Anonymous requests do not require that the client prove its identity. If a client attempts to bind directly as an LDAP object and make a query, the LDAP server can accept or reject the query, based on ACI (Access Control Information).

In this chapter, Sun ONE Directory Server 5.1 will be used as the Directory Server. This software package can be downloaded from Sun (as described later). The Sun ONE Directory Server includes a Java-based Console and the Administration Server. The Console is a Java-based GUI front end for the Sun ONE Directory Server. The Administration Server handles requests from Console clients and translates those requests into the appropriate configuration changes or data manipulation in the Directory.

A final note about terminology. Some of Sun's product names are currently in transition. Originally, the LDAP server was known as the *iPlanet Directory* server, but under Sun it has been officially renamed *Sun ONE Directory Server*. Most documentation still refers to the LDAP server under the old name of iPlanet. Since Sun intends to move to the name "Sun ONE" for all iPlanet products, we will refer to these products as *Sun ONE* when we are talking about Sun's product line, but we will generically refer to the *Directory Server*, the *Administration Server* and the *Console*, where appropriate.

Where Did LDAP Come From?

To better understand how an LDAP server and client are implemented under Solaris 9, it is useful to have a clear understanding of the origins of directories and the methods used to access them.

Work on meeting the need to replace paper directories with an international standard for electronic directory storage was worked on by the CCITT (Consultative Committee on International Telegraphy and Telephones), now known as the International Telecommunications Union, ISO (International Standards Organization) and the ECMA (European Computer Manufacturers Association) between 1984 and 1993. What emerged was the X.500 directory standard, which arranged information into objects, object classes that objects can belong to, and attributes that objects can have if they are in particular object classes. What made this interesting was the fact that new object classes could be created that were very similar to existing object classes with only slight additions or modifications. (OOP programmers will find this concept familiar.) This made it very easy to store standard information in a standard way, and to store site-specific or implementation-specific information in an easily extensible way.

Before LDAP, DAP (Directory Access Protocol) was used to access X.500 directories. Unfortunately, DAP was a very “meaty” OSI protocol stack that was difficult to fit onto a small client system. The client side required considerable resources to access directory information. This excluded the rather weak workstations that were available at the time that DAP was created. In practice, only other servers could be DAP clients. LDAP was developed by the University of Michigan so that workstations could access X.500 directories with less overhead.

LDAP is very similar to DAP in that it uses X.500’s data and namespace model. LDAP differs from DAP in that it was designed to work directly on the TCP/IP protocol stack. LDAP also uses a simple string encoding to carry data, rather than DAP’s more complex data encoding methods. LDAP also uses free reference implementations available from the University of Michigan. For these reasons, it has been much simpler for developers to create applications that speak LDAP.

Originally, LDAP was designed only to be a protocol to let *lightweight* clients access X.500 directories. In practice, however, features like replication, enhanced security and full-featured management tools have come together and many people use the word “LDAP” to mean the protocol, the directory, and the infrastructure surrounding them.

The Directory Server

The Sun ONE Directory Server 5.1 is a full-featured directory that speaks both LDAP version 2 and LDAP version 3. The Directory Server, the Administration Server and the Console can all be installed on one computer, or the Console can be placed on a separate workstation. Officially, the Directory Server that ships with Solaris 9 is the only contract-supported Directory Server from Sun. The author of this chapter has set up the downloaded Directory Server on Solaris 9 02/02 without any problems at all. Try working with the included version of the Directory Server. This will reduce your setup time and will generally work seamlessly with the default install of Solaris 9. If for some reason the included server does not work for your needs, download and install the full standalone 5.1 version of the Sun ONE Directory Server.

The Console is a Java-based tool that can work on almost any platform with the proper JRE (Java Runtime Environment). The Java version that comes with Solaris 9 02/02 works just fine with the Directory Server. The Java Runtime Environment can be downloaded from <http://www.sun.com/> in the download section.

Understand there are two different types of Java packages. One package is the JDK (Java Development Kit) that is used by developers. The other Java package is the JRE (Java Runtime Environment) that is used by Java clients. LDAP requires the JRE.

Where to Download the LDAP Server

To download a trial copy of the Sun ONE Directory Server (full version):

1. Browse to <http://www.sun.com/downloads>
2. Browse to the “Internet and Identity Management” section.
3. Select the Sun ONE Directory Server 5.1. The downloaded file will probably be called something like
4. **directory-5.1-us.sparc-sun-solaris2.8.tar.gz**

This file has a size of 51 MB. Only readers with high speed Internet access should try to download this file.

Documentation on the Sun ONE Directory Server can be found at:

<http://docs.sun.com/db?p=prod/sunone>

At this writing, the URL for the Directory 5.1 documentation was

http://docs.sun.com/db?p=coll/S1_ipDirectoryServer_51

Just as a reminder, the names of some of Sun’s products are currently in transition. Even though the *iPlanet* product line has been officially renamed *Sun ONE*, most of the documentation still refers to it as *iPlanet*.

The current Directory Server 5.1 documentation includes:

- Administrator's Guide
- Configuration, Command, and File Reference
- Deployment Guide
- Installation Guide
- Schema Reference
- SP1 Release Notes
- Release Notes

These guides can be downloaded in HTML or PDF format. HTML files require the use of a standard Web browser. The PDF files require the use of the Adobe Acrobat Reader.

Requirements for the Sun ONE Directory Server include:

- **Memory:** A minimum of 256 MB. However, 1024MB is the recommended minimum memory for an LDAP server in a production environment. If a system has less than 256 MB of memory, the LDAP server will not run.
- **Free disk space:** A minimum of 200 MB. Sun recommends 2 GB (2048 MB) of disk space for production servers. The author of this book has installed the Directory Server on a Sun Blade 100 without any major problems.

The main components of the Directory Server are:

Directory Server	A fully functional LDAP v3 and v2 server. The server runs with a daemon named ns-slapd .
Console	A Java-based console that can be run from the server itself or from a satellite workstation over a network. The console can be used with the Directory Server or any other supported Sun ONE/iPlanet server. The console can be used to start, stop and manage servers from the network. This package includes the Directory Server and the Administration Server.

Administration Server This server receives all communications from all Consoles, processes the requests, and then sends them to the appropriate server (in this case, the Directory server.)

Help Files The help files are packaged in HTML format. They can be viewed with any modern Web browser. Adobe Acrobat help files are not included in the download package.

Required DNS Server Setup

The Directory Server needs to resolve the fully qualified domain name of the workstation. A DNS server must be able to respond to give back an IP address to a <hostname>.<domain> inquiry.

If the command

```
ping yourhostname.yourdomain.com
```

produces the response

```
yourhostname.yourdomain.com is alive
```

then a DNS server is resolving the fully qualified domain name (hostname.domain) for the test workstation, and the Directory Server will work. Otherwise, see Chapter 29 (the DNS chapter) for instructions on how to set up DNS on your system.

Lesson 30.1 Configuring the Directory Server Included with Solaris 9

In this lesson, we will configure the Directory Server included with Solaris 9, using the Custom Install option. This installation will also install the Administration Server and the Console, as well as the documentation that accompanies the Directory Server, the Administration server and the Console.

This lesson is based on a Solaris 9 workstation with the following settings:

Hostname	= sun100
IP Address	= 192.168.0.4
Subnet Mask	= 255.255.255.0
Domain	= aassddff.com
Default router	=192.168.0.1
DNS server	= 192.168.0.1
DNS server	= < Any_Other_DNS_Server_IP_Address >

It's best to use the above settings for this lesson. You can use your actual network specific information, but other chapter lessons will have to be modified accordingly to work properly.

This lesson also assumes that you can successfully ping your test workstation, using either the command **ping sun100.aassddff.com** or **ping < hostname.domain >**. Make sure that the **ping** command works before you start this lesson.

Note that the default configuration is a simple LDAP Master Server. It is possible to create a master/replica pair for redundancy in case the main LDAP server goes down. It is also possible to create a Multi-Master setup with several LDAP servers dividing up the LDAP DIT (Directory Information Tree). However, in this lesson, all data is on one Directory Server that is being set up on a single workstation.

1. Log in as the root user.
2. Open a Terminal window.

3. Create an appropriate group and user for the Directory Server to use. Type the following commands:

```
groupadd -g 389 sunone  
useradd -u 389 -g sunone -c "Directory Server user" ids
```

It is common for UNIX-based server applications to operate under the identity of a particular user account.

The group ID used here (389) is the same as the TCP port that listens for LDAP requests. A number of UNIX operating systems have adopted this convention for daemons that listen on well-known ports because it's an easy way to avoid collision—the ports are guaranteed to be unique for each service. Don't worry about setting a password or creating a profile for this user, as the password will be locked by default so that it cannot log in interactively.

*Note also that the username here is **ids** (iPlanet Directory Server) and the group name is **sunone**. If you install any other Sun ONE products on this machine, it's common to have them run as different users that all share the same group name. For example, if you installed Sun ONE Messaging Server on the same machine, it could run as a user called **ims** that belongs to the **sunone** group.*

4. Run the **idsktune** utility to evaluate your system's suitability for running Directory Server:

```
/usr/iplanet/ds5/bin/slapd/server/idsktune
```

*Attend to any significant issues that are listed.
There are three types of records in the output:*

ERROR	<i>The Directory Server will not run if the problem is not repaired.</i>
WARNING	<i>This problem will not prevent the LDAP server from running, but it might significantly degrade the server's performance.</i>
NOTICE	<i>This warning describes best practices for setting up an LDAP server.</i>

For example, on one test system, the author received the following message (among others):

```
WARNING: Only 512MB of physical memory is available on the system.  
1024MB is the recommended minimum.
```

This is properly classified as a WARNING. If a system had less than 256 MB of RAM, this an ERROR would be generated.

Note: Many NOTICES have to do with TCP tuning parameters. Two good resources for tuning these parameters are:

- *An excellent automatic way to tune a number of these parameters to more sane and secure values is a Bourne shell script which is maintained by Jens-S. Vöckler of the University of Hanover. It can be found at <http://www.sean.de/Solaris/nettune>. The author of this chapter installs **nettune** on most production systems that he maintains.*
- *Some detailed and useful information about tuning the Solaris TCP/IP stack can be found at <http://www.sean.de/Solaris/soltune.html>.*

5. Start the configuration process with this command:
/usr/sbin/directoryserver setup

6. *The first screen just shows some basic navigation information:*

Sun-Netscape Alliance

iPlanet Server Products Configuration

Welcome to the iPlanet Server Products configuration program
This program will configure iPlanet Server Products and the
iPlanet Console on your computer.

You must have "root" privilege to configure the
software.

Tips for using the configuration program:

- Press "Enter" to choose the default and go to the next screen
- Type "Control-B" to go back to the previous screen
- Type "Control-C" to cancel the configuration program
- You can enter multiple items using commas to separate them.
For example: 1, 2, 3

Would you like to continue with configuration? [Yes]: <Enter>

Press Enter to accept the default and continue with the configuration.

7. *The next screen lets you install the Console and Directory Server, or just the Console alone.*

Select the items you would like to configure:

1. iPlanet Servers

Configures iPlanet Servers with the integrated iPlanet Console
onto your computer.

2. iPlanet Console

Configures iPlanet Console
as a stand-alone Java application on your computer.

Select the component you want to configure [1]: <Enter>

Press Enter to accept the default: [1] iPlanet Servers.

8. *You will see the following options:*

Choose a configuration type:

1. Express Configuration

Allows you to quickly configure the servers using the most
common options and pre-defined defaults. Useful for quick
evaluation of the products.

2. Typical Configuration

Allows you to specify common defaults and options.

3. Custom Configuration

Allows you to specify more advanced options. This is
recommended for experienced server administrators only.

Choose a configuration type [2]: 3 <Enter>

Choose option 3 (Custom Configuration). *This will illustrate more of the configuration options so that you can use them if needed:*

9. *The next screen gives a listing of the server products.*

iPlanet Server Products components:

Components with a number in () contain additional subcomponents which you can select using subsequent screens.

- 1. iPlanet Directory Suite (2)**
- 2. Administration Services (2)**

Specify the components you wish to configure [All]: <Enter>

Press Enter to select all the components.

10. *Because the Custom Installation option was selected earlier, the next screen shows the components available with the iPlanet Directory components.*

iPlanet Directory Suite components:

Components with a number in () contain additional subcomponents which you can select using subsequent screens.

- 1. iPlanet Directory Server**
- 2. iPlanet Directory Server Console**

Specify the components you wish to configure [1, 2]: <Enter>

Press Enter to select all the components.

11. *The next screen shows the Administrative Components*

Administration Services components:

Components with a number in () contain additional subcomponents which you can select using subsequent screens.

- 1. iPlanet Administration Server**
- 2. Administration Server Console**

Specify the components you wish to configure [1, 2]: <Enter>

Press Enter to install both the Administration Server and the Console.

12. *The next screen asks for the fully qualified domain name (FQDN) of the server:*

**Enter the fully qualified domain name of the computer on which you're configuring server software. Using the form <hostname>.<domainname>
Example: eros.airius.com.**

Computer name [sun100.aassdff.com]: <Enter>

Enter your hostname followed by the same domain name in the `/etc/defaultdomain` file.

If your FQDN doesn't show up between the square brackets [] abort the configuration by pressing CTL + C. Then, fix the problem with your DNS configuration and then start over. In the experience of the author, the configuration script will only work if the Fully Qualified Domain Name can be resolved.

13. *The next screen asks for the user and group that will be used to run the Directory Server.*

Choose a Unix user and group to represent the iPlanet server in the user directory. The iPlanet server will run as this user. It is recommended that this user should have no privileges in the computer network system. The Administration Server will give this group some permissions in the server root to perform server-specific operations.

If you have not yet created a user and group for the iPlanet server, create this user and group using your native UNIX system utilities.

**System User [nobody]: ids <Enter>
System Group [nobody]: sunone <Enter>**

Enter **ids** as the system user, and **sunone** as the system group.

14. *The next question only applies to a situation in which another Directory Server has been installed and this server's configuration is stored on another server:*

iPlanet server information is stored in the iPlanet configuration directory server, which you may have already set up. If so, you should configure this server to be managed by the configuration server. To do so, the following information about the configuration server is required: the fully qualified host name of the form <hostname>.<domainname>(e.g. hostname.domain.com), the port number, the suffix, and the DN and password of a user having permission to write the configuration information, usually the iPlanet configuration directory administrator.

If you want to install this software as a standalone server, or if you want this instance to serve as your iPlanet configuration directory server, press Enter.

Do you want to register this software with an existing iPlanet configuration directory server? [No]: <Enter>

In this case, this is a standalone server, so press Enter to select **No**.

15. *The next screen asks if this server will store user and group information:*

If you already have a directory server you want to use to store your data, such as user and group information, answer Yes to the following question. You will be prompted for the host, port, suffix, and bind DN to use for that directory server.

If you want this directory server to store your data, answer No.

Do you want to use another directory to store your data? [No]: <Enter>

Press Enter to select No.

Some highly secure networks have a separate LDAP server for sensitive information like passwords and HR information, but in this lesson, we are installing everything onto a single server..

16. Select the TCP port that LDAP will use:

The standard directory server network port number is 389. However, if you are not logged as the superuser, or port 389 is in use, the default value will be a random unused port number greater than 1024. If you want to use port 389, make sure that you are logged in as the superuser, that port 389 is not in use, and that you run the admin server as the superuser.

Directory server network port [389]: <Enter>

Press Enter to select the default. The default port number for LDAP across most architectures is **389**.

If you get an error stating that that port is not available, you may have another instance of the Directory Server already running. Verify this as follows:

- a. Open a new Terminal window.
- b. Enter one of the following commands, to find TCP connections that may be using the port:
`netstat -an | grep "\.389"`
or
`netstat -a | grep ldap`

If you find any matches, you'll need to do some research to figure out why the port is being used, and then turn it off. Look for possible LDAP processes in the process table with the command `ps -ef`. The most likely problem is a previous installation.

17. The next screen asks for the Directory server identification name. The default is the hostname of the system.

Each instance of a directory server requires a unique identifier. Press Enter to accept the default, or type in another name and press Enter.

Directory server identifier [sun100]: <Enter>

Press Enter for the default.

18. The next screen asks for the administrator ID for this installation of your Directory Server:

Please enter the administrator ID for the iPlanet configuration directory server. This is the ID typically used to log in to the console. You will also be prompted for the password.

**iPlanet configuration directory server
administrator ID [admin]: <Enter>
Password: [output suppressed]admin123 <Enter>**

Password (again): [output suppressed] admin 123 <Enter>

- a. Press Enter to accept the default administrator ID, which is admin.
- b. Then type admin123 as the password both times you are asked for it.
The password typed in will not be shown on the screen.

*Note that this should **not** be a user ID found in the `/etc/passwd` file. This is a user created within the directory itself.*

19. The next screen asks for the LDAP suffix:

The suffix is the root of your directory tree. You may have more than one suffix.

Suffix [dc=aassdff, dc=com]:

Select the defaults.

If the defaults don't look something like `dc=<domain name>`, `dc=com` then you may have DNS name resolution issue that should be resolved before you continue.

20. Next, a Directory Manager's name must be selected:

Certain directory server operations require an administrative user. This user is referred to as the Directory Manager and typically has a bind Distinguished Name (DN) of `cn=Directory Manager`. Press Enter to accept the default value, or enter another DN. In either case, you will be prompted for the password for this user. The password must be at least 8 characters long.

Directory Manager DN [cn=Directory Manager]: <Enter>

Password: [output suppressed] direct123 <Enter>

Password (again): [output suppressed] direct123<Enter>

- a. Press Enter to select the default name of `cn=Directory Manager` or
If a different directory manager name is desired, type `cn=< username >`.
- b. When prompted for the password, type direct123

Understand that the Directory Manager is as powerful within the directory as the root user is on the server.

21. The next screen is used to select the administration domain:

The Administration Domain is a part of the configuration directory server used to store information about iPlanet software. If you are managing multiple software releases at the same time, or managing information about multiple domains, you may use the Administration Domain to keep them separate.

If you are not using administrative domains, press Enter to select the default. Otherwise, enter some descriptive, unique name for the administration domain, such as the name of the organization responsible for managing the domain.

Administration Domain [aassddff.com]: <Enter>

Press Enter to accept the system's domain name as the default.

22. *The next screen asks for sample entries.*

You may install some sample entries in this directory instance. These entries will be installed in a separate suffix and will not interfere with the normal operation of the directory server.

Do you want to install the sample entries? [No]: yes <Enter>

Type **yes** so that the sample entries are added to the system.
These sample entries will be used in later lessons.

23. *The next question asks if the system has an LDIF file to populate the LDAP database.*

You may wish to populate your new directory instance with some data. You may already have a file in LDIF format to use or some suggested entries can be added. If you want to import entries from an LDIF file, you may type in the full path and filename at the prompt. If you want the install program to add the suggested entries, type the word **suggest at the prompt. The suggested entries are common container entries under your specified suffix, such as **ou=People** and **ou=Groups**, which are commonly used to hold the entries for the persons and groups in your organization. If you do not want to add any of these entries, type the word **none** at the prompt.**

Type the full path and filename, the word **suggest, or the word **none** [suggest]: none <Enter>**

This file does not exist now, so just type **none** and continue.

24. *The next screen asks about schema checking:*

If you are going to import an old database immediately after or during installation, and you think you may have problems with your old schema, you may want to turn off schema checking until after the import. If you choose to do this, schema checking will remain off until you manually turn it back on. iPlanet recommends that you turn it back on as soon as possible.

Do you want to disable schema checking? [No]: <Enter>

Press Enter to select **No**
(In this case a previous database is not going to be imported.)

25. *The next screen asks for the administration port:*

The Administration Server is separate from any of your application servers since it listens to a different port and access to it is restricted.

Pick a port number between 1024 and 65535 to run your Administration

Server on. You should NOT use a port number which you plan to run an application server on, rather, select a number which you will remember and which will not be used for anything else.

The default in brackets was randomly selected from the available ports on your system. To accept the default, press return.

Administration port [25791]: <Enter>

Press Enter to accept the default (the default shown above is selected randomly during each install and will probably be different from yours).

Make a note of this default, because this is the port that the iPlanet Console will need to know so that it can connect to the directory.

26. *The next screen asks what IP address should the Administration Server bind to:*

If you want to configure the Administration Server to bind to a specific IP address, enter the address below.

To accept the default shown in brackets, press the Return key.

IP address []: <Enter>

Do not type in an IP address; just press the Enter key.
This will make the Directory Server listen on all interfaces by default.

27. *The Administration Server needs to run as a specific user:*

The Administration Server program runs as a certain user on your system. This user should be different than the one which your application servers run as. Only the user you select will be able to write to your configuration files. If you run the Administration Server as "root", you will be able to use the Server Administration screen to start and stop your application servers.

Run Administration Server as [root]: <Enter>

Press Enter to accept the default (root).

*While it is possible to run the Directory Server daemon (**ns-slapd**) as a user other than root, this will mean that the process cannot listen on the default port (389) and must listen on a non-privileged port (1024 or higher). However, running a server daemon as root has inherent security problems. Some future version of the Directory Server will probably segment off and secure this in some way.*

28. *The configuration process will now begin:*

```
[slapd-sun100]: starting up server ...  
[slapd-sun100]: [08/Sep/2002:12:28:00 -0800] - iPlanet-Directory/5.1 B2002.057.0855 starting up  
[slapd-sun100]: [08/Sep/2002:12:28:05 -0800] - slapd started. Listening on all interfaces port 389 for  
LDAP requests  
Your new directory server has been started.  
Created new Directory Server  
Start Slapd Starting Slapd server configuration.  
Success Slapd Added Directory Server information to Configuration Server.
```

Configuring Administration Server...

Your parameters are now entered into the Administration Server database, and the Administration Server will be started.

Changing ownership to admin user root...

Setting up Administration Server Instance...

Configuring Administration Tasks in Directory Server...

Configuring Global Parameters in Directory Server...

iPlanet-WebServer-Enterprise/6.0SP1 B11/09/2001 13:59

warning: daemon is running as super-user

[LS ls1] http://sun100.aassdff.com, port 25791 ready to accept requests

startup: server started successfully

Press Return to continue...

Go to /usr/sbin and type `directoryserver startconsole` to begin managing your servers.

When prompted, press the Enter (Return) key.

Note: Some of the lines above have been wrapped to fit the page.

29. Type the command `/usr/sbin/startconsole`

After the command `/usr/sbin/startconsole` is typed the following screen appears:

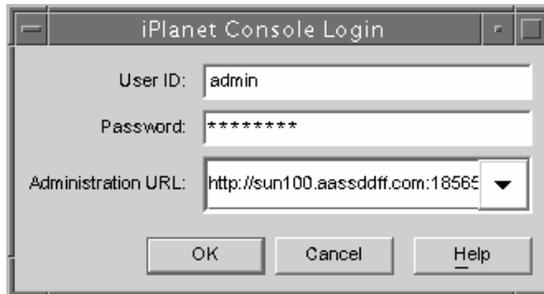


Figure 30.1 iPlanet Console Login

Type in the password `admin123` and left click **OK**.

In a production environment, you would obviously choose a stronger password.

30. Now you will see the iPlanet Console, which is a GUI front end to the Netscape Directory Server

You can play around with settings here, but careful! Make sure to press the Cancel button at any screen that you modify.

Future lessons in this chapter will show you how to actually work with the Netscape Directory Server.

31. To gracefully exit from the iPlanet Console:

- a. Left click on the Console menu bar item.
- b. Left click on Exit.

The Directory Console

The Directory Console (also called the *iPlanet Console*) is the GUI front end to the entire family of Sun ONE/iPlanet servers. Figure 30.2 is a snapshot of the Console window.

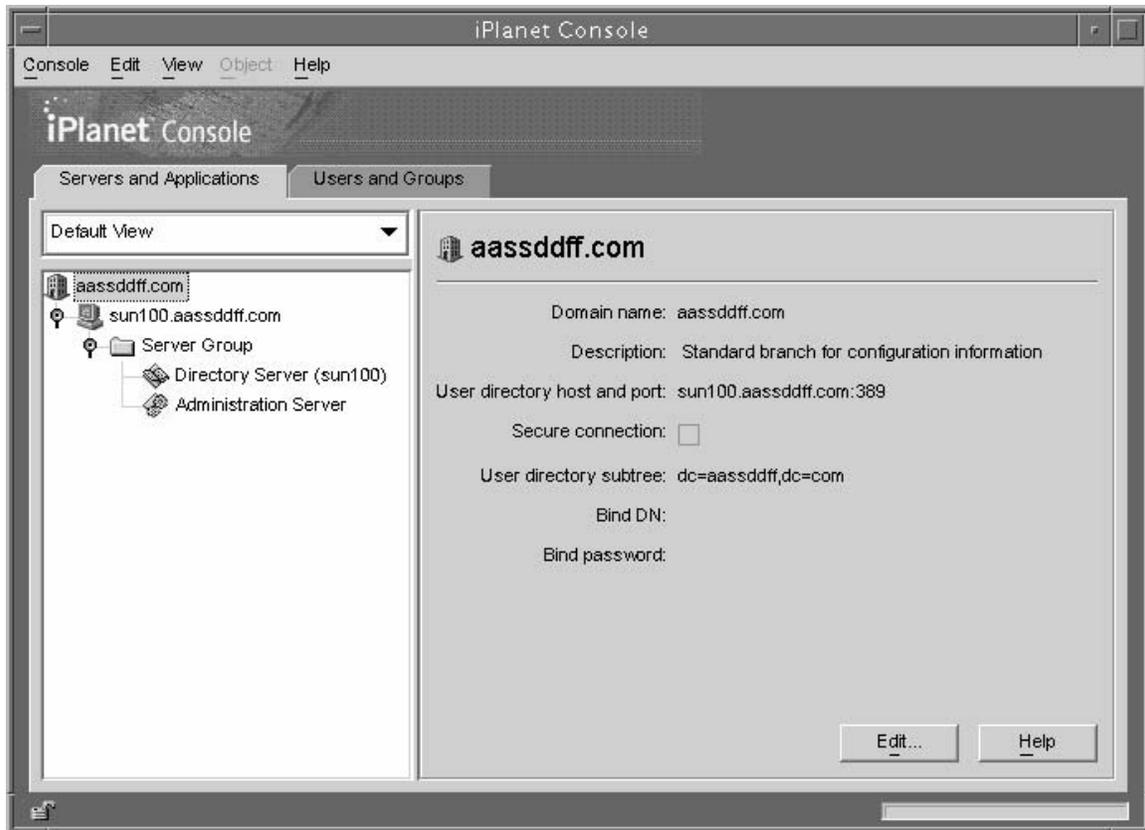


Figure 30.2 The Directory Console Window

On the very top of the Console window is a menu bar.



Figure 30.3 iPlanet Menu Bar

The menu bar changes dynamically depending on the highlighted object. When the Console first starts, it has the menu choices: Console, Edit, View, Object and Help. The menus that result drop down from the menu bar also dynamically change, depending on the object selected. For example, the Console drop down menu has the choice Create Administration Domain when the console is first started. When the Administration Server is started, The Console drop down menu has the selection Security as the first item. Under the menu bar is a banner. The banner is the blue picture on the left side that reads “iPlanet Console.” When the Console is being run, the banner is displayed as “iPlanet Console.”



Figure 30.4 iPlanet Banner

This banner changes, depending on what Sun ONE/iPlanet server is being run. When the Administration Server is running, the banner changes to “iPlanet Administration Server”.

The status bar at the bottom right of the screen works just like the status bar on an HTML browser. It displays information such as provides locations and busy messages.



Figure 30.5 iPlanet Status Bar

Under the banner are two tabs: Servers and Applications, and Users and Groups. Let's look at each in turn.

The Servers and Applications Tab



Figure 30.6 Servers and Applications

When the Servers and Applications tab is selected, the left View pane shows the Administration Domain that the console is currently working inside. In this example, it should be aassdfff.com.

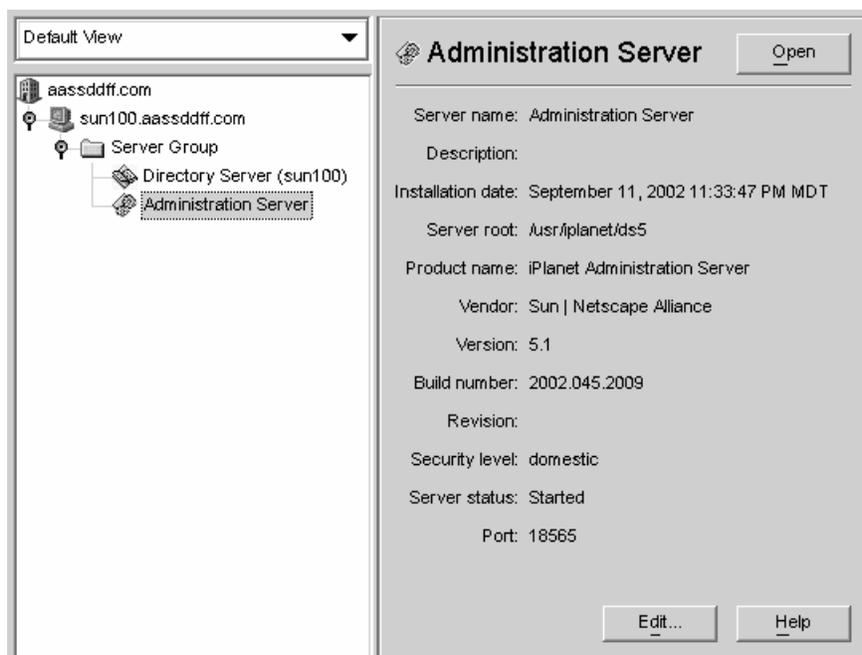


Figure 30.7 Administration View Pane

The Right information pane shows detailed information on the object that is highlighted in the left View pane.

There is a small button on the bottom right corner labeled “Edit...” This button can be used to change the server name and description shown in the left View pane. This only changes the server’s display name in the Console. It does not actually change the server’s host name. It is a rather harmless way of organizing servers. Most companies change the display name to something more useful like “Dallas-Admin”.



Figure 30.8 Information Pane

The Help button, also on the lower right corner only displays information on the Edit... button. For the server’s official documentation, left click on the Help menu item on the Menu bar.

The Users and Groups Tab

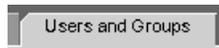


Figure 30.9 Users and Groups

The other tab is the Users and Groups tab. Clicking on this calls up a window that is used to search for users or groups in the directory. This window can also be used to create, modify or delete a user.

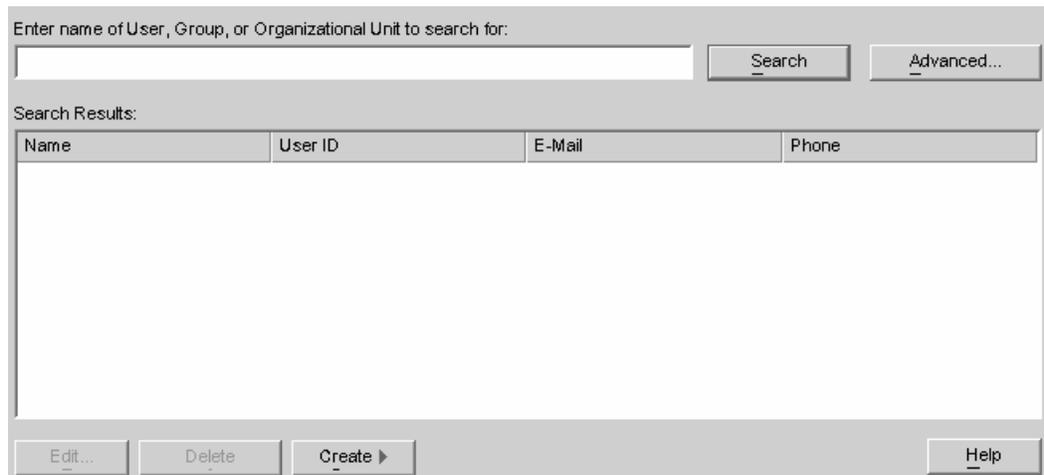


Figure 30.10 Users and Groups Window

Lesson 30.2 Working with the iPlanet Console

The first window that is displayed is the Console Login window. This window lets the system administrator choose which Directory Server to connect to.

In this lesson, we will start the Administration Server from the command line. After the Administration Server is started, we will start the Console (the front end GUI for the Administration Server and the Directory Server). We will simply familiarize ourselves with the Console without changing anything and then will exit the Console.

Note: While the Console is useful for general server configuration, most large-scale production environments do not use the Console for user provisioning. Previous versions of the Console have been buggy and semi-unpredictable. Instead, many larger organizations usually write in-house applications to write user provisioning information to the directory.

1. Log in as the root user.
2. Open a Terminal window.
3. Start the Directory server (if it is not running):
`/usr/sbin/directoryserver start`
You should see a message like `/usr/planet/ds5/slapd-sun100/start-slapd`

Note: If the server is already running, you will see a message like
`There is an ns-slapd process already running: 667.`
Ignore this message.

4. Start the Administration Server (if it is not running):
`/usr/sbin/directoryserver start-admin`
There should be a message that includes lines like
`http://sun100.aassdfff.com , port 25791 ready to accept requests.`
`startup: server started successfully`
The hostname and port displayed on the message will vary.

Note: If the server is already running, you will see a message like
`/usr/iplanet/ds5/start-admin`
`iPlanet-WebServer-Enterprise/6.0SP1 B11/09/2001 13:59`
Ignore this message.

5. Type the command `/usr/sbin/directoryserver startconsole &`
If a message appears saying “cannot contact the LDAP server on port 25791, Would you like to attempt to restart the Directory Server”:
 - a. Cancel out of the Console login screen.
 - b. Start the LDAP server with the `/usr/sbin/directoryserver start` command.
 - c. Just to be safe, type the command `/usr/sbin/directoryserver start-admin` to make sure the iPlanet Administration Server is running.

The command `/usr/sbin/directoryserver startconsole &` command starts the Console in the background. The Console contacts the Administration Server. The Administration Server is a collection point for all the instances of the Console. It passes all the Console requests to the various Sun ONE/iPlanet servers. The Directory Server is one of many different types of servers that can take requests.

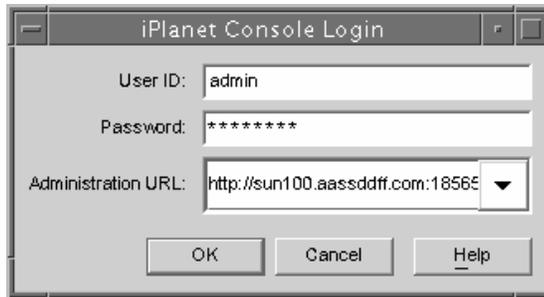


Figure 30.11 iPlanet Console Login

6. Log in as the User ID **admin**
*The UserID requested is for the administrator of the Administration Server. Ordinary users can also log in to this screen. For this example, log in as **admin**.*
*The password is **admin123** (or whatever you used during configuration)*
*The Administration URL: should be **http://sun100.aassddff.com:25791/**.*
7. In the left View pane, left click on the Turner icon (the magnifying glass with a + sign) to the left of the **sun100.aassddff.com domain**.
A folder labeled "Server Group" should appear. In this example, there are only two servers available: the Administration Server and the Directory Server.
8. Click on the Administration Server.
The right pane now shows information on the Administration Server.

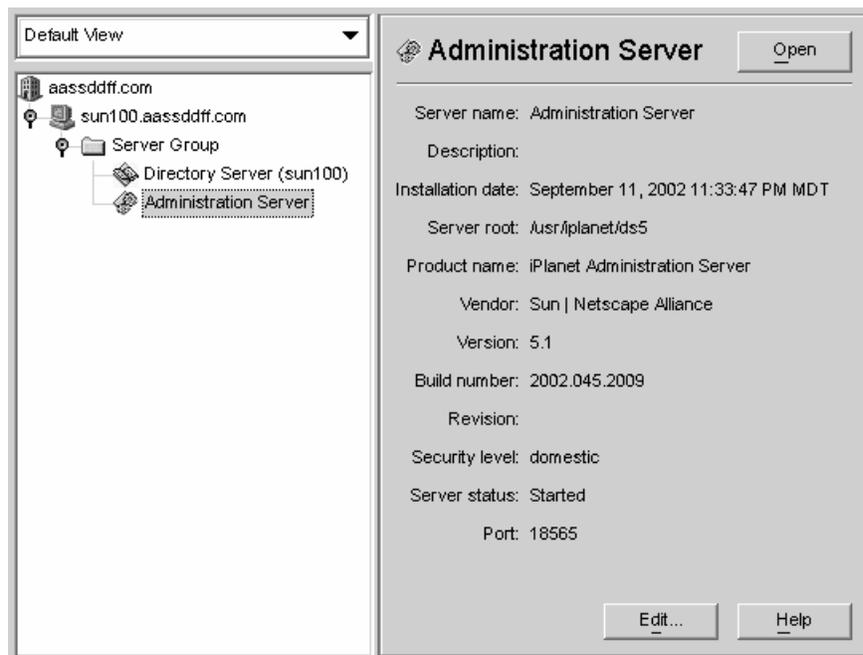


Figure 30.12 Administration View Pane

9. Left click on the Administration Server icon in the left View pane.
10. Left click on the Open button at the top right corner.
*This should open a window titled "Administration Server" The banner now says **iPlanet Administration Server**.*

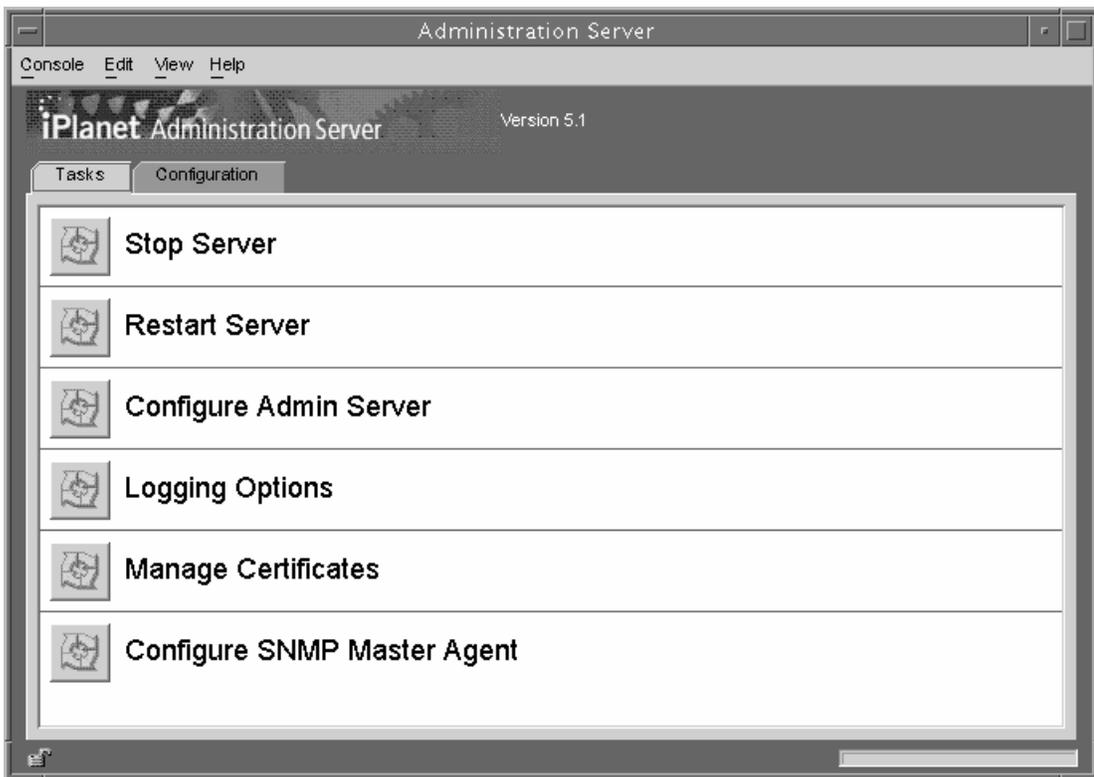


Figure 30.13 Administration Server Window

11. Now close the Administration Server window by double-clicking on the Window Menu button. It looks like a minus sign on the top left of the window frame.
The point of this lesson is not to work with the Administration Server, but to explore the Console Server. This window is only being shown as an example.
12. Left click on the Directory Server.
13. Left click on the Open button in the top right corner of the window. The following window appears:

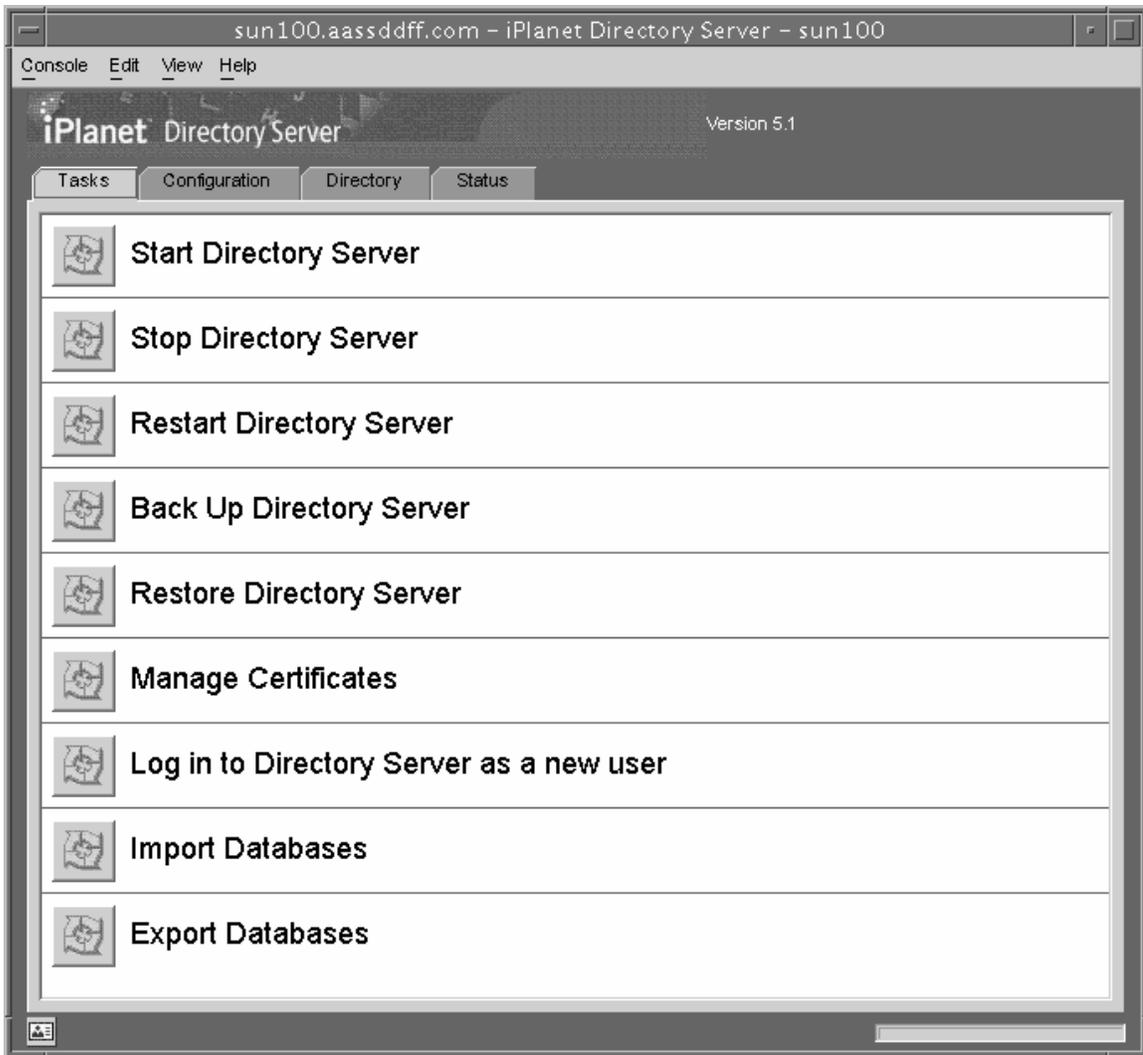


Figure 30.14 Directory Server Window

14. Close the Directory Server window by double-clicking on the Window Menu button..
15. Left click on Edit in the Menu bar.
16. Left click on Preferences...
17. Left click on the Restore Defaults button.
18. Left click on OK.
The Restore Defaults menu item should return the console back to its original display settings. This feature is somewhat buggy and doesn't appear to work consistently.
19. Left click on View in the Menu bar.
20. Left click on Banner Bar, Status Bar and Tree a few times.
21. Left click on the Console menu bar item.
22. Left click on Exit.

Working with Users and the Administration Account

The **admin** user that was created during the initial configuration has two roles:

- Configuration Administrator

In the LDAP tree this user has the following characteristics: uid=userID, o=NetscapeRoot, ou=TopologyManagement and ou=Administrators. The Configuration Administrator is used to modify the configuration directory.

- Administration Server Administrator

This user can perform functions such as managing the Administration Server, starting and stopping the server, making configuration changes, and modifying the security policy. This user exists only in the file `/usr/iplanet/ds5/admin-server/config/admpw`. This file has the administrator's name and encrypted password.

The startconsole Command

The `startconsole` command is used to start the Console. The `startconsole` command takes the following options:

- a** HTTP location of the Administration Server. For example:
`http://sun100.aassdff.com:25471`
- f** `<error_file>` Send errors to a text file.
- h** Display a simple terminal help message.
- l** `<language>` Use a different language than the default language.
- u** `<username>` Start the Console with a given username, most likely **admin**.
- w** `<password>` The password for the user. This is a dangerous item to put in a script. If a hacker cracked this, the company LDAP system could be jeopardized.
- x** `<nologo,nowinpos>` Do not display the Sun ONE splash screen; position the login screen at the top left.

Lesson 30.3 Creating a Console Startup Script and Changing the Console View

In this lesson we start the Console with a custom Bourne shell script. After the Console is started, the right View pane will be customized so only the servers that the administrator needs to work with are displayed.

This lesson assumes that the Directory Server and the Administration Server are both running. If they are not running, perform steps 3-4 of Lesson 30.2.

Note that this script contains sensitive password information. In a production environment, this script would be created with security in mind and would not include this information.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command `vi startc`
Add the following two lines to the script.
`#!/bin/sh`
`/usr/iplanet/ds5/startconsole -u admin -w admin123 -a http://sun100.aassdff.com:25471`
4. Save this Bourne shell script with the name `startc`
5. Type the command `chmod 777 startc`
Because this is a test workstation, full read/write/execute permissions can be set on this script. In a production environment, the script's permissions and ownership would be more restrictive, usually something like `chmod 700 startc`.

6. Run the script by typing the command **./startc**
*In a production environment, a script like this might be prohibited, because it has a clear text password.
 In a production environment, check the company's official policy on automated login scripts. Alternative ways of proceeding include logging in with the Console Login window, and using a similar script that does not use the **-w** option.*
7. Log in as the User ID: **admin** with your password.
*The Administration URL: should be **http://sun100.aassdff.com:25471**
 The port number could be different, depending on the port chosen during the installation.*
8. Left click on the View menu bar choice.
9. Left click on Custom View Configuration...

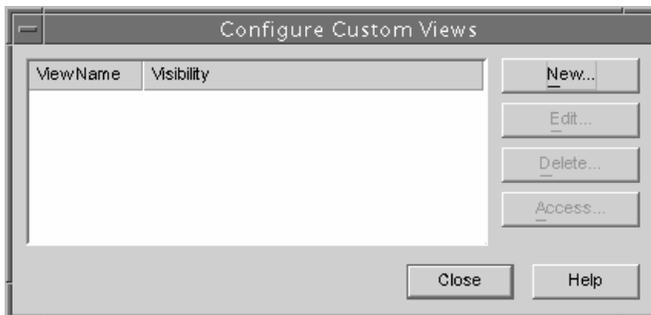


Figure 30.15 Configuration Views Window

Once the Console window has started, it is possible to change what items are shown in the left View pane. Instead of seeing all the administration domains and servers, the View can be changed to only show servers that apply to a particular LDAP administrator.

10. Left click on New... (in the window shown above). The following window appears:

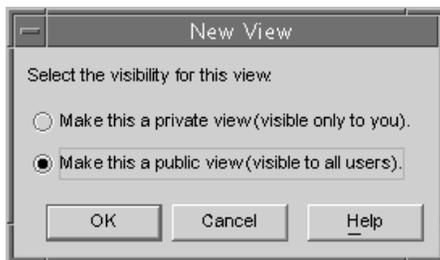


Figure 30.16 New View Window

11. Check the lower radio button with the title “Make this a public view (visible to all users).”
12. Left click on the OK button. *The Edit View window appears.*

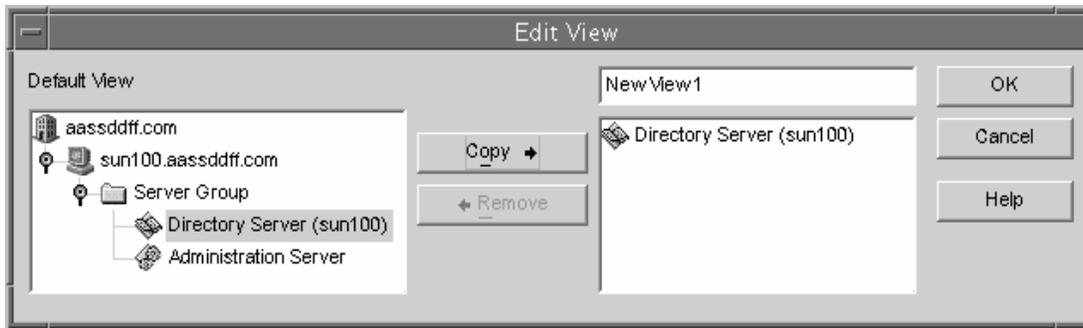


Figure 30.17 Edit View Window

13. In the Edit View window, expand the tree in the left View pane until the “Directory Server” icon is visible, then click on that icon it to highlight it (the result is shown above).
14. Left click on the Copy → button to move the icon to the right View pane (the result is shown above).
15. Change the name of the view (above the right view pane) from NewView1 to DirectOnly.
16. Left click on the OK button.
17. Left click on the Close button.
18. Above the right View pane is a drop down menu with the text “Default View.” Left click the down arrow that is next to the text.
19. Select the DirectOnly view that you just created.
*As shown in the image on the right, this view shows only Directory Servers.
 In a large company with hundreds of Sun ONE Servers, it is convenient to have a customized view that only shows the servers that you specify.*
20. Left click on the Console menu bar item.
21. Left click on Exit.



Lesson 30.4 Creating a Configuration Administrator with Console

The first administrator created was the admin user. This administrator was created during the installation of the Directory Server. Understand that this administrator is only a Configuration Administrator, a userid that works on the Console, the Administration Server, and the Domain Server.

This lesson shows how to create another administrator. Most large companies require everyone who works on a server to have an individual account. This is necessary for system security and audit purposes.

This lesson assumes that the Console is already running.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command `/usr/sbin/directoryserver startconsole &`
If a message appears saying cannot contact the LDAP server on port 25791, Would you like to attempt to restart the Directory Server:
 - a. Cancel out of the Console login screen.
 - b. Start the LDAP server with the `/usr/sbin/directoryserver start` command.
 - c. Just to be safe, type the command `/usr/sbin/directoryserver start-admin` to make sure the iPlanet Administration Server is running.

4. Left click on the Users and Groups tab.
5. Left click on the **Advanced...** button.
*The **Advanced...** button is located on the top right side of the window.
 The following window should appear:*

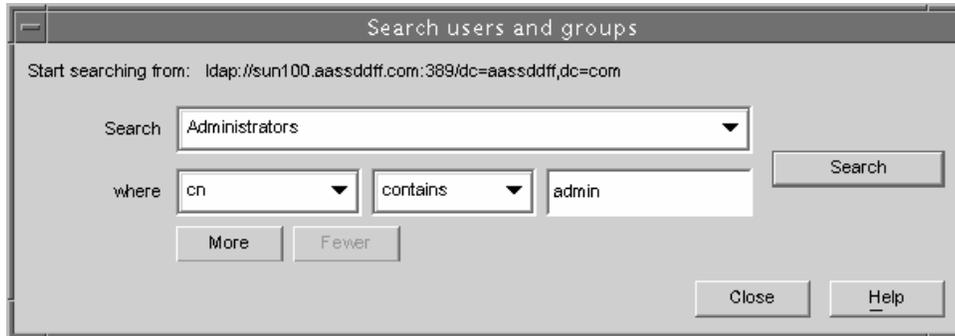
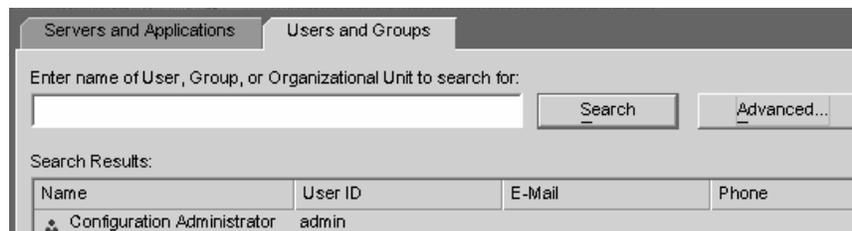


Figure 30.18 Search Users and Groups Window

6. In the Search field, select **Administrators**.
*The following three steps refer to the row of text fields entitled **where**.*
7. In the left field, select **cn**.
8. In the center field, select **contains**.
9. In the right field type in **admin**
The window should now look as shown above.
10. Left click on the Search button.
11. Left click on the Close button.
Now the Users and Groups tab should have the Configuration Administrator highlighted, a shown below.



12. Left click on the **Edit...** button in the lower left corner of the window (not shown above).
13. *Browse through the administrator's information. Ignore NT User and Posix User information—they do not apply to Sun Solaris. When finished browsing, click on the Cancel button. Do not change this use; if the settings are incorrect, there will be no administration user to work with.*
14. Left click on the **Cancel** button
15. Left click on the **Create ►** button
16. Left click on the **Administrator...** menu choice

Fill in the form with the following pieces of information:

First Name: **admin2**
Last Name: **admin2**
Common Name(s): **admin2 admin2**
User ID: **admin2**
Password **myadmin2pw**

Confirm Password **myadmin2pw**

In a production environment, you would obviously choose a stronger password.

The screen now should look as shown below.

The screenshot shows a 'Create Administrator' dialog box. The title bar reads 'Create Administrator'. The main window has a sidebar on the left with 'User' selected. The main area contains the following fields:

- * First Name: admin2
- * Last Name: admin2
- * Common Name(s): admin2 admin2
- User ID: aadmin2
- * Password: *
- * Confirm Password: *
- E-Mail: (with a placeholder example: user@company.com)
- Phone:
- Fax:

At the bottom of the main area, it says '* Indicates a required field'. At the bottom of the window, there are four buttons: 'Access Permissions Help', 'OK', 'Cancel', and 'Help'.

Figure 30.19 Create Administrator Window

17. Left click on the OK button.

If you now perform the same search for an administrator as in step 6 of this lesson, two names should be displayed: admin and admin2..

When using the admin2 account, use the public MyArea view that created in the previous lesson.

18. Left click on the Console menu bar item.

19. Left click on Exit.

Working with the Administration Server

The Administration Server is accessed through the Console. The Administration Server is used to collect all the information from the Consoles and redirect that information to the various Sun ONE Servers.

To access the Administration Server:

1. In the left View pane of the Console window (shown below), highlight the Administration Server icon.
2. Left double click the Administration Server icon, *or* click the Open button.



Figure 30.20 iPlanet Console

A new window will open, with two tabs:

The Tasks Tab

There are six icons are under this tab. These tabs are:

- | | |
|-------------------------------|--|
| Stop Server | Stops the Administration server. Generally, it may be safer to use the command line utilities to stop the server. |
| Restart Server | Stops and then starts the Administration Server. This is necessary if certain types of changes are made to the Administration Server. It may be safer to use the stop-admin and start-admin command line utilities instead of this icon. |
| Configure Admin Server | Calls up a popup window with five tabs. |
| Network | Port#, IP address, Server UID, Connection Restrictions based upon domain, IP or hostname. This allows you to lock down who is allowed to administer the directory, and from where. |
| Access | The administrator's username and password, as well as if the Directory server has access to the Administration Server. |
| Encryption | If SSL 2.0 or SSL 3.0 (Secure Sockets Layer) is required to contact the Administration Server. This supports the RSA cipher family. |
| Configure DS | If the Administration Server is going to use a different Directory Server, the LDAP hostname and LDAP port number would be entered here. |
| User DS | Only used if the user's account is not on the local LDAP server. The user's home LDAP server, port, and other foreign LDAP server information can be used on this page. |
| Logging Options | Used to set the location of the error and access log files. Basically, this is where the Administration Server log files are kept. |

There are two text log files:

Error Log	Saves different types of messages, such as warning messages, error messages, warnings, notices and non-critical events.
Access Log	Used for security purposes. Indicates who accessed the server at what time and from what location.
Manage Certificates	Deals with Digital Certificates, such as what certificates are active, passwords on the certificate and the CA authority.
Configure SNMP Master Agent	SNMP (Simple Network Management Protocol) is a method of network monitoring where the network device or server sends out vital statistics about itself to an SNMP server, such as HP OpenView. The Administration Server uses port 161 by default. GET and SET SNMP parameters can be added from this screen. The Manager is the HP OpenView or SNMP server used to monitor the Administration Server. If an SNMP server is not present, there is no reason to use this icon.

The Configuration Tab

The Configuration tab basically performs some of the same tasks that were displayed with the icon under the Tasks tab. This window shows the LDAP directory that the Administration Server is using in the directory. In the right View pane is a partial listing of the LDAP tree that concerns this Administration Server.

Administration Server icon	Gives the same configuration information as the Configure Admin Server icon under the Tasks tab.
SNMP Master Agent	Provides the same SNMP configuration as under the Configure SNMP Master Agent icon under the Tasks tab.
Logs	Provides the same log file maintenance window that was seen with the Logging Options icon under the Tasks tab. The only difference here is that the Error Log file and the Access Log file can be viewed under this icon. It is much easier to just use the <code>vi</code> editor in read only mode and look at the log files directly. The viewing window is rather small and hard to use.

Files Related to the Administration Server

One of the best ways to learn about a new piece of software is to browse the directory that holds that software. The Sun ONE servers are located in the `/usr/iplanet/ds5` directory. The official documentation is a good place to start, but it's often very educational to look at the files themselves and become familiar with them.

Key Files Related to the Administration Server

- The `/usr/iplanet/ds5/adminacl` directory has several key LDAP files:

`generated.https-admserv.acl` has the following text:

```
Version 3.0;
acl "admin-defaults";
authenticate (user, group) {
```

```

prompt = "webServer Server";
};
allow (read, list, execute,info) user = "anyone";
allow (write, delete) user = "all";

```

This file defines the authentication used with the LDAP server. The file is really self explanatory.

- The `/usr/iplanet/ds5/admin-server/config` directory has the following files:

adm.conf This file also shows the following pieces of information.

```

LdapHost:      sun100.aassddff.com ← Location of the default LDAP server
LdapPort:     389 ← Port the LDAP server is using
cn=sun100.aassddff.com ← Domain of the LDAP server
siepid         password ← Password for the Admin server's administrator
port:         25471 ← Port the Administration Server is using
ldapStart:    slapd-sun100/start-slapd ← Command used to try and start the LDAP
server

```

admpw This file contains the names of the administrator user and an encrypted password. If the administrator's password is lost, you can manually lift an encrypted password that you do know from a UNIX system and place it here, as long as that system uses the same encryption method. If a Sun ONE user's name is not in this file, the user is not an administrator.

local.conf This provides several key pieces of information

```

serverroot: /usr/iplanet/ds5           Default location for the Sun ONE servers on
this system

configuration.nsServerPort: 25471     Port that the Admin Server listens on.

configuration.nsErrorLog:admin-serv/logs/error Location of the error log

configuration.nsErrorLog:admin-serv/logs/pid   Location of the text file that shows the
PID of the Admin server

configuration.nsErrorLog:admin-serv/logs/access The location of the access log

```

- The `/usr/iplanet/ds5/admin-serv/logs` directory has three log files used by the Administration Server:

error This text file shows the startup and shutdown of the Administration Server. Figure 30.21 is a sample screen shot of the error file:

```

[08/Sep/2002:04:13:23 -0800] - Backing up file 38 (/usr/iplanet/ds5/slapd-
sun100/bak/2002_09_08_04_13_19/log.0000000013)
[08/Sep/2002:04:13:25 -0800] - Backing up file 39 (/usr/iplanet/ds5/slapd-
sun100/bak/2002_09_08_04_13_19/DBVERSION)
[08/Sep/2002:04:13:25 -0800] - Backup finished.
[08/Sep/2002:04:55:27 -0800] - Beginning export of 'NetscapeRoot'
[08/Sep/2002:04:55:27 -0800] - export NetscapeRoot: Processed 107 entries (100%).
[08/Sep/2002:04:55:27 -0800] - Beginning export of 'sampleRoot'
[08/Sep/2002:04:55:27 -0800] - export sampleRoot: Processed 160 entries (100%).
[08/Sep/2002:04:55:27 -0800] - Beginning export of 'userRoot'

```

[08/Sep/2002:04:55:27 -0800] - export userRoot: Processed 2 entries (100%).
 [08/Sep/2002:04:55:27 -0800] - Export finished.

Figure 30.21 Sample Lines from the Error Text File

access When the Administration Server is accessed, it keeps a log of each connection. This text file shows where a request came from (data such as hostname, IP, domain, time and date) and what type of request was attempted. This is useful for troubleshooting connections, identifying possible intruders, auditing and developing applications.

pid The PID (Process IDentification number) of the Administration Server. If for some reason the Administration Server will not shut down properly, this PID can be used with the **pkill** or **kill** commands to shut the server down.

Another configuration file is the file
`/usr/iplanet/admserv5.1/admin-serv/config/magnus.conf`

Some useful information to pull out of this file is: shown below:

DNS	on/off	Shows if DNS is being used
Security	on/off	Shows if security is being used
ClientLanguage	en	Shows the default language of the Console
AdminLanguage	en	Shows the default language of the Administration Server
DefaultLanguage	en	Shows the default Administration Server language

Lesson 30.5 Administration Server Basics

This lesson introduces the reader to some of the most basic configuration changes possible with the Administration Server. These tasks include granting access to the server based upon a hostname and IP address, and viewing the **error** and **access** log files.

This lesson assumes that the Directory Server and the Administration Servers are running.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command `/usr/sbin/directoryserver startconsole &`
4. Log in as the User ID: **admin** and enter your password.
5. In the left view pane, left click on the Turner icon (+ sign) next to the aassddff.com administrative domain.



This could be a different domain, depending on the setup of your test system.

6. Left click on the Turner icon next to the host sun100.aassddff.com
This could be different depending on your system's hostname.
7. Left click on the Turner icon next to the Server Group.
8. Left click on the Administration Server icon.
9. Left click on the Open... button in the Information pane on the right.
There are two tabs on the Administration Server window.
10. Left click on the Tasks tab.



11. OPTIONAL : Left click on the Restart Server icon.
You might want to restart the server if you are not sure that changes that you have just made are taking effect. It could take about three or four minutes for the Administration Server to actually restart. A small popup window should appear with the message

Issuing restart request..

Restart request accepted by the server.

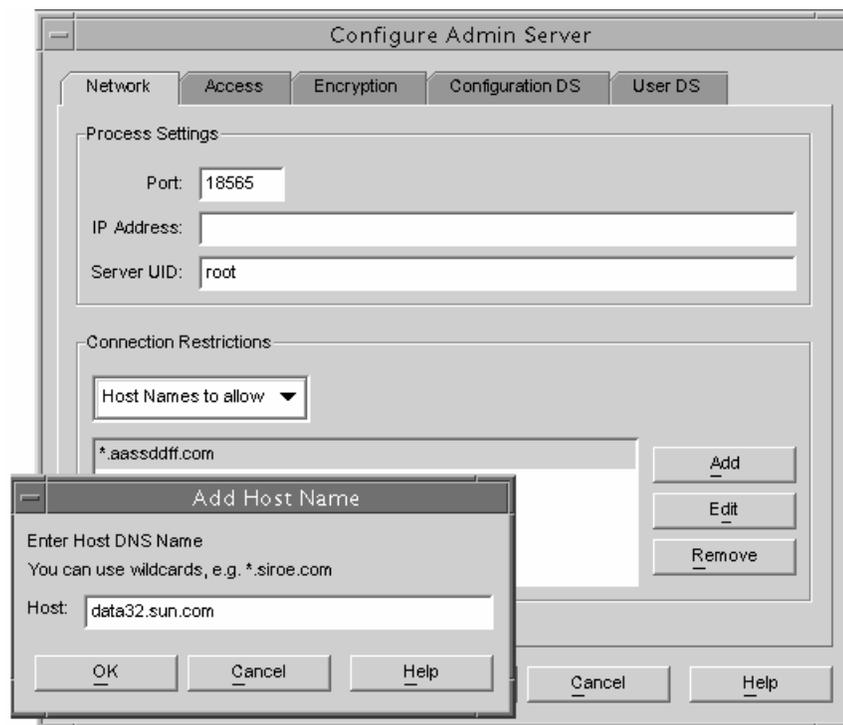
Waiting for the server to restart...

If the Administration Server does not restart in an appropriate amount of time, follow these steps:

1. Left click on the Cancel button.
2. Left click on the Console Menu bar item
3. Left click on the Exit menu choice.
4. Open a Terminal Window.
5. Type the command
`/usr/sbin/directoryserver start-admin`
This restarts the Administration Server by hand.
6. Type the command `startconsole &`

12. Left click on the Configure Admin Server tab.
 13. Left click on the Network tab.

14. Left click on Add at the bottom of the screen.
 15. In the Host: field, type **data32.sun.com**
This illustrates how to restrict access to this server to only the IP addresses or hostnames of the systems that belong to the administration group in a company. In the default configuration, any host in the default domain is allowed to access the Administration Server.



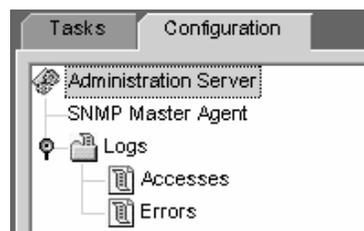
16. Left click on the OK button
Be wary of the other tabs (Access, Encryption, Configuration DS, and User DS) and do not experiment with their options unless you are fully aware of their effects.
17. Left click on the OK button on the Configure Admin Server window to close the window.
At this point, you should be back to the original Admin Server window.

Logging Options

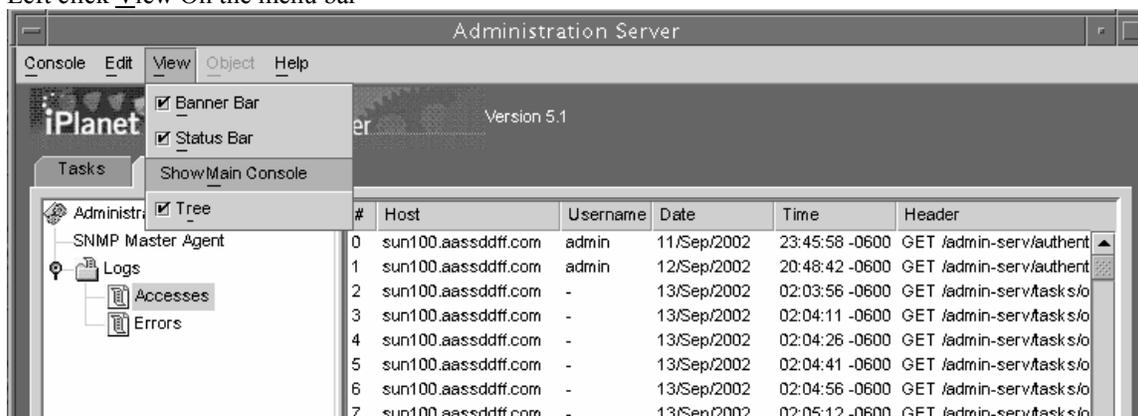
18. Left click on the Logging Options icon.
The popup window shows the location of the two log files. They are located in the directory /usr/iplanet/ds5/admin-serv/logs. The root directory /usr/iplanet/ds5 is not included in this path—it is a relative path.

19. Left click on the Cancel button. 

20. Left click on the Configuration Tab.
21. Left click on the Turner icon to the left of the Logs icon.
22. Left click on the Accesses icon.
This is one of the two log files associated with the Administration Server. As mentioned earlier, it is much easier to open a Terminal window and use a command line text editor than to use this GUI to examine the error logs.



23. Left click View On the menu bar



There are four items on the View drop down menu, Left click on the Show Main Console menu choice. The Show Main Console menu item displays a small icon of the original Console window. This can be double left clicked to return to the original Console Window.

24. Left click on Console menu bar item
25. Left click on Exit menu bar item.
This will exit the Console. Note that the Console choice Close only closes this window. It does not exit the Console.

The Sun ONE Directory Server

The Sun ONE Directory Server is a modern LDAP v2 and v3 compliant server. This server was originally designed for Solaris 8. The server is installed on Solaris 9 02/02 even though it was not originally created for Solaris 9. If a newer version of the Sun ONE Directory Server is created for Solaris 9, use that version instead of the version referenced in this book.

Directory Basics

Before using the Directory Server it would be a good idea to understand what an LDAP directory is and how it is used in a corporate environment. An LDAP directory is basically a collection of information known as the DIT (Directory Information Tree). LDAP-based naming services use trees and subtrees in the directory, just as NIS used maps and NIS+ used tables.

Some directory trees are modeled after units in an organization. This could be an Internet domain name like aassdff.com or a department like Sales or Marketing. A very high traffic directory is the user directory. This directory holds information on all the users in a company. Most companies also have a Directory Suffix that matches the company's domain, as in dc=aassdff,dc=com. (There is a special entry in the DIT tree that has the distinguished name (DN) "Directory Suffix." This entry represents the DIT root.)

Sun ONE servers often store their own configuration data within the directory itself. Most of this information can be found in the o=iPlanet tree.

It is possible to use what is known as an Attribute Map to map a service to another name, just as a person's name can be attached to a Social Security number. Attribute maps can look like:

```
attributemap: group:GID=Group Numbers
attributemap: passwd:Comment=Real_Name
```

The term *schema* refers to the directory's plan for relating information to other information, permitting particular attributes to be attached to particular objects, and governing what kind of data an attribute can hold. Many of the schema guidelines followed by many Directory Servers (including the Sun ONE /iPlanet Directory Server) are inherited from the basic schema standards governed by the IETF (Internet Engineering Task Force). For more information go to the <http://www.ietf.org/> web site.

The *default schema* is the schema expected by the client, with components such as ou=hosts or ou=user. Software developers could not develop LDAP-aware applications if one LDAP server used the term ou=userid and another LDAP server used the term ou=user, so there are default schemas that are used throughout the industry. Many of these were defined as part of the original X.500 specification.

Schemas are covered in more detail below.

Types of LDAP Directory Servers

If an organization has only one LDAP server, that server is the Master Server by default. This server accepts all client read and write requests. If there is too much LDAP traffic hitting one Directory Server, a company can create a duplicate of the directory's data in another directory using *replication*.

There are a number of ways to arrange replication agreements between Directory Servers:

- | | |
|---------------------------|--|
| Multi-Master Servers | In this environment, there are two master Directory Servers. Each server can perform all the typical LDAP functions like read, writes and searches. This can be a problem when it comes time to synchronize the directories between Directory Servers. The best policy is to have separate directories being handled by each server. If both masters receive updates to a particular object, the most recent change is considered valid. |
| Single Master Replication | With this model, a Master Server and a Replica Server exist. The Master Server can perform all client LDAP read, bind, compare and write requests. The Replica Server can not perform write operations. Instead, it refers any write operations to the master. This eliminates the possibility of having conflicting LDAP entries in the DIT. |

However, if the Master Server goes down, the Replica Server cannot perform write operations on its own.

Floating-Master Server This model is very similar to the Single Master Replication model. When both the Master Server and the replica are up, the Master Server is the only server that can perform write operations. The Replica Server can perform all LDAP functions except writing to the DIT. If the Master Server goes down, the Replica Server is promoted to a Master Server. The newly created master writes LDAP information to the DIT. When the original master is brought back online, the converted master updates the information on the master and then demotes itself back down to a Replica Server.

LDAP Terms

Before discussing LDAP servers and LDAP clients, it might be a good idea to become familiar with some common LDAP terms.

Configuration Directory Administrator	The default name for this account is <i>admin</i> . This account can be thought of as a super user for Sun ONE Servers. When someone logs in with this account, he or she can perform administration on any Sun ONE Server.
Administration Server Manager	This account can perform administration tasks on the Administration Server. This account is usually the same as the Configuration Directory Administrator's account (the admin account).
Directory Manager	The default name for this account is "Directory Manager." It can be changed in the Configuration tab of the Directory Server GUI. The Directory Manager is an account that can be thought of as a root user for LDAP. This account can bypass any authorization and pretty much do anything desired. The password of the Directory Manager must be at least eight characters in length.
LDAP entry	An LDAP entry is any record in the LDAP database. An LDAP entry is composed of a <i>Distinguished Name</i> (explained later) and a collection of one or more attributes and values: for example : uid = asmith.
LDIF	LDAP Data Interchange Format. This is an RFC-based plain-text format to exchange data between directories. Data can be exported as an LDIF, manually edited, and then imported again. Data extracted from other sources can be reorganized into an LDIF and then imported into the directory. LDIFs usually have an .ldif extension.
Distinguished Name	A Distinguished Name is a special attribute located in an LDAP entry. This attribute is referenced with the abbreviation <i>DN</i> . The DN is the "primary key" of sorts for each object in the directory. Every object has a DN, and each full DN is always unique.
LDAP Tree	The tree that holds all the company information. The LDAP tree's root directory is just like the root directory of the Solaris file system. It branches out to things like country, organizations, corporate divisions, departments, individuals, and office telephone numbers.
SSD	Service Search Descriptors. SSDs are used to tell the LDAP naming service client where and how to search for a particular service.

LDAP v3

LDAP version 3 is defined by the IETF in RFC 2251. Some significant features include:

- LDAP can operate over the SSL (Secure Sockets Layer) and SASL (Simple Authentication and Security Layer) frameworks.
- Intelligent Referrals. This feature makes it look to users as if there is a single directory for information, even if the sources of the information come from scattered servers. These scattered servers can be local or they can be contacted over the Internet. This lets users perform Internet-wide address book lookups.
- Support for International Character Sets. LDAPv3 supports UTF-8 encoding and language tags. LDAP administrators can now allow users to view LDAP information in their multiple languages (even in the same window). LDAP v3 supports more than 38 languages. This is useful for multi-national companies that have a centralized LDAP presence.
- Dynamical Extensible Schema. Applications can now write user customization data to the directory. This can be such things such as user preferences, shared data and configuration data. This can be useful for “roaming users” such as sales people, who connect remotely to the network through a VPN (Virtual Private Network).
- Protocol Extensibility. New operations can be added to the protocol.

TLS Transport Layer Security. This refers to the ability of the LDAP client and the LDAP server to have secure communications. The most common type of security is SSL (Secure Sockets Layer) communications.

PAM Pluggable Authentication Methods. This is the implementation of the pam_ldap or pam_unix security model.

Directory Server HTML Documentation

When the Directory Server is installed on a system, it comes with HTML documentation. This documentation can be read by any modern HTML browser.

The HTML help manuals for the Administration Server and Console are found in the following directories:

/usr/iplanet/ds5/manual/en/admin/ag/contents.htm

Table of Contents for the HTML webpages related to the Console and Administration Server

/usr/iplanet/ds5/manual/en/admin/ag/ix.htm

Index of terms related to the Console and Administration Server

/usr/iplanet/ds5/manual/en/admin/ag/glossary.htm

Glossary of terms used by the Console and Administration Server

/usr/iplanet/ds5/manual/en/console/help/login.htm

This is the HTML help screen that is shown when a user clicks on the “HELP” button. It is a good idea to modify this HTML page with company-specific information.

/usr/iplanet/ds5/manual/en/slaped/index.htm

This is the homepage of the Directory Server documentation. This includes the following guides:

- Deployment Guide
- Administrator’s Guide

- Configuration, Command and File Reference
- Schema Reference

This webpage also has links to the more extensive documentation online at Sun's web site <http://docs.sun.com/>.

Introduction to the Directory Server

The Directory Server is accessed through the Console. There are four tabs in the Directory Server window: Tasks, Configuration, Directory, and Status. These tabs are discussed below.

The Tasks Tab

This tab displays a series of icons that allow the system administrator to accomplish simple tasks with just a click of the mouse. Basically, this is a shortcut for some of the most complex tasks that need to be performed on an LDAP server.



Start Directory Server

This icon is used to start the Directory Server if it is not running. If the server is running, this will not harm the server. A message appears saying **The server <hostname> is already running**. If for some reason the Directory Server does not start, leave the Console and then type the commands

```
cd /usr/iplanet/ds5
cd slapd-<hostname>
./start-slapd
```

Stop Directory Server

This icon is used to stop the Directory Server. When this icon is clicked, a warning message appears saying **Are you sure you want to stop Directory Server <hostname>?** If you click on the Yes button, the Directory Server is stopped. If for some reason the Directory Server does not stop properly, exit the Console and type the commands

```
cd /usr/iplanet/ds5
cd slapd-<hostname>
./stop-slapd
```

Restart Directory Server

Most of the time, when a change is made in the LDAP directory tree, the changes do not require a restart of the LDAP server. However, when changes are made to the configuration of the actual Directory Server itself, a restart is sometimes required. If you choose this and the LDAP server does not start again, use the commands **stop-slapd** and **start-slapd** as described above to manually stop and start the LDAP server.

Back Up Directory Server

This is a very useful option that creates a full backup of the database in the specified directory. The backup directory is named for its creation time:

```
/usr/iplanet/ds5/slapd-<hostname>/bak/YEAR-MONTH-DAY-HHMMSS
```

The backup directory has two files and three directories.

The two files are:

DBVERSION Text file with one line “Sun ONE-ldbm/6.0”
log.0000000xxx Binary file for transaction logging information

The three directories match the LDAP directories, and contain **.db3** files:

NetscapeRoot
sampleRoot
userRoot

Restore Directory Server This performs the exact opposite function of the Back Up Directory Server icon—it restores the directory from a specified set of backup files.

Manage Certificates This deals with Certificates of Authority and digital certificates. This is used with some encryption methods. This window lets users request, install and revoke certificates from a Certificate Authority.

Log in to Directory Server as a new user

This lets the Console User access the Directory Server as a new user, provided that he or she is in the following LDAP structure:

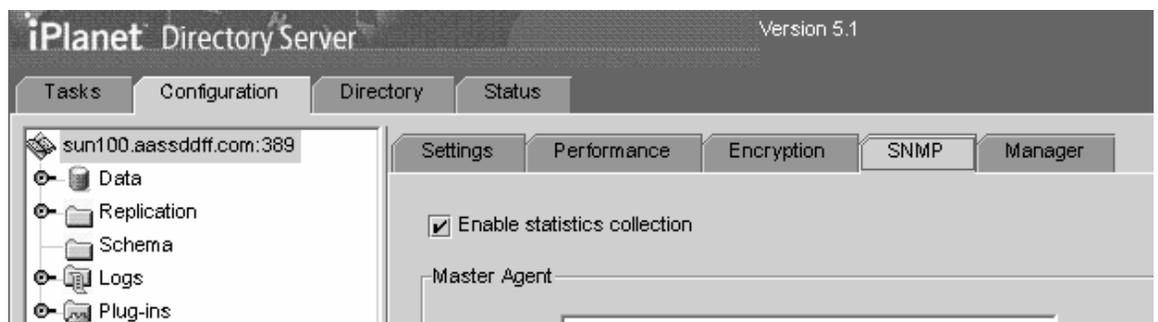
uid=<userid>
ou=Administrators
ou=Topology Management
o=NetscapeRoot

A password must also be supplied.

Import Database This icon lets the administrator import an LDIF file. The LDIF file is a text version of data that can be stored in the directory. This utility can be used to import an entire directory, an entire subtree, or a single entry directory.

Export Database This icon lets the administrator export an LDIF file. This utility can be used to save an entire database or just a subtree of the LDAP tree. The file should have a file name extension of **.ldif**

The Configuration Tab



Unfortunately (or fortunately, depending on your perspective), there are so many administration functions that can be performed in the Configuration tab that it would take about 200 pages to describe everything. Only a few key administration tasks will be discussed here. Most of this information is easy to spot if you just browse through

the different tabs and highlight the different elements in the left pane. This information is most useful for readers who don't have a SPARC workstation but would like to get some basic familiarity with the LDAP server.

sun100.aassdfff.com

When the server itself is highlighted, the server's settings can be changed. There are five tabs that are displayed in the right View pane.

Settings Tab

Port

If you elect not to use the standard LDAP ports (389 or the 636 Secure LDAP port), these port settings can be changed here.

Make entire server read-only

This is a good option to use for a public LDAP server. If a public LDAP server needs maintenance, disable LDAP client access and then uncheck this box. Perform the LDAP maintenance and then enable the "read-only" mode. Finally, allow LDAP clients to use the server again. During certain operations, such as building indexes, the directory will place itself in this mode. In some older versions of the Directory Server, it may occasionally leave itself in this mode. You can usually uncheck it here when the configuration operation has finished.

The track entry modification times

This option makes the Directory Server create and/or modify attributes called modifyTimestamp and createTimestamp. This is a little bit of additional overhead but can be very handy when troubleshooting a problem with individual entries in the directory or when tracking down which entries were accidentally modified during a bulk change.

Enable Schema checking Schema checking enforces rules about which attributes are allowed by which object classes. This should always be enabled. Entries need to follow the schema rules that apply to their creation. It would make no sense for a password object to have a telephone number.

Performance Tab

This tab has entries that limit the server's thresholds for the number of LDAP entries, time limit for a request, and the idle timeout. Be very careful when modifying these parameters. Clients could suffer unreasonable timeout errors, or the server could not function at full capability if these settings are incorrect. Only experienced LDAP administrators should change these settings.

Encryption Tab

This tab changes the security settings on the LDAP server by requiring clients to connect with different types of SSL authentication. The most important thing to realize is that different encryption schemes should be tried one at a time. Don't re-configure the server with five or six new encryption schemes at once. Make a simple change on a test server and then see what happens. Then try the other encryption schemes and check performance. Experimentation coupled with some research can help you develop a solid security model that works well for you.

SNMP Tab

Unless you are monitoring your network with something like HP OpenView, there is no need to work with these settings. SNMP sends heartbeat and health information to an SNMP server. The SNMP server then alerts an administrator if something requires attention. Unless you are actively using SNMP, it is unneeded overhead and can be a security risk.

Manager Tab

This controls the name and password of the Directory Server manager. Be very careful when working with this setting—you can lock yourself out of the directory.

Icons in the Left View Pane

The following icons appear in the left View pane when you have selected the Configuration tab.

- Replication** This is used to allow the directory to replicate its data to other Directory Servers (be a *supplier*) or to allow the Directory Server to receive replication data from another server (be a *consumer*). This can be used to provide redundancy or to distribute processing load.
- Schema** The schema contains all the rules that govern what data is allowed in the directory and how the data relates to other data in the directory. In this area, the administrator can create a new LDAP object class. The object name, parent and OID (Object Identifier) can be specified, as well as the object class's required attributes and allowed attributes. The Machine Rules can be observed, but not modified, in the Schema area.
- Logs** When this icon is highlighted, the three standard log files associated with the Directory Server can be viewed. The Creation and Deletion policies control how often a log file is created (for example, every day, every two days, every week) and how large the log file can become before it can not be written to anymore. The deletion policy on a log file determines when the log file should be deleted, depending on the log file's age, the amount of free disk space, and the size of the saved log files.
- The three log files are:
- Access log** The access log shows the LDAP requests and the Directory's responses to the requests. It shows the date and time and connection to the LDAP server, as well as a very detailed information sheet. Note that the client's IP address is shown in this log file when the client attaches to the LDAP server. The default location for the access log is
`/usr/iplanet/ds5/slaped-sun100/logs/access`
- Error log** The error log saves warning messages, error messages and other server significant messages. This includes backups and restores. The default location for the error log is:
`/usr/iplanet/ds5/slaped-sun100/logs/errors`
- Audit Log** The audit log shows the changes made to each database and the server's overall health. The location for the audit log is
`/usr/iplanet/ds5/slaped-sun100/logs/audit`
- Plug-ins** A plug-in can be thought of as a small piece of software that lets the LDAP server perform new functions, without the need to recompile the server itself. Plug-in files usually have an `.so` file extension. This screen lets the administrator enable or disable a plug-in.

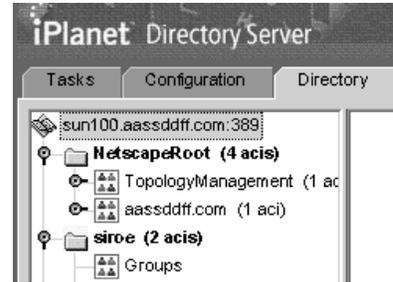
The Directory Tab

This is the LDAP directory. All objects in the LDAP tree are here, including user data, configuration data and schema data.

Some Key Directory Subtrees

Some important subtrees in the Directory tab are:

- siroe** This is the example data that is included with the installation of the Directory Server. Feel free to experiment with the data in this subtree.
- NetscapeRoot** This is a subtree that is used by the Console, the Directory Server and the Administration Server. This shows the Users, Groups and Administrators that are inside the Console Manager. This subtree also holds Administration Server configuration data. Try to avoid managing this directory structure directly here; stick to using the GUI.
- config** This is the configuration data for the Directory Server itself. This subtree should not be manually edited unless absolutely necessary. Try to use the configuration GUI instead.



The Status Tab



The Status Tab is a great tool to use when LDAP users complain that the Directory Server is too slow. This shows the resources used, and the requests that the Directory Server is handling. If a Directory Server is taking too many hits, the data needs to be replicated to another server.

Rather than go through each item, we will review some significant key points.

- In the Performance Counters section you will find the startup time of the server. If there was a report that the LDAP server was not available, this is the easiest place to check the LDAP server's start time and uptime.
- The checkbox [] **Continuous refresh** should be checked when trying to troubleshoot directory issues that are changing in real-time. It gives an update about every 5 seconds. Avoid using this unless necessary, as it places more load on the Directory Server.
- In the Resource Summary box, a number of statistics about use of the Directory are available. Focus on the most immediately useful statistics at first:
 - Connections, Average per Minute: how many clients are connecting.
 - Operations Completed, Average per Minute: how many operations were successful. It may be useful to compare this value to Operations Initiated.
 - Bytes sent to Clients, Average per Minute : shows how much data is being sent.
- In the Global Database Cache Information, the Hit Ratio is probably the most useful. A high value (near 100) means that the cache is performing efficiently.

Lesson 30.6 LDAP Save, Restore, Export and Import

In this lesson readers will save the contents of the LDAP server to a directory. This directory contains the LDAP data files and index files. Later, the LDAP server will be restored from the same directory. Finally, readers will export the LDAP DIT tree to an LDIF file.

This lesson assumes that the Directory and Administration servers are running.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command `/usr/sbin/directoryserver startconsole &`
*If a message appears saying **cannot contact the LDAP server on port 25471, Would you like to attempt to restart the Directory Server**, cancel out of the Console login screen and start the LDAP server with the **start-slaped** command. This is described earlier in the chapter.*
4. Log in as the admin user, with your password.
5. In the left View pane, click on the Turner icon next to the aassddff.com administrative domain.
6. Left click on the plus Turner icon next to the host sun100.aassddff.com
7. Left click on the plus Turner icon next to the Server Group.
8. Left click on the Directory Server (sun100) icon.
9. Left click on the Open button located in the top left corner of the right-hand screen.
10. Left click on the Tasks tab.
11. Left click on the Back Up Directory Server icon.

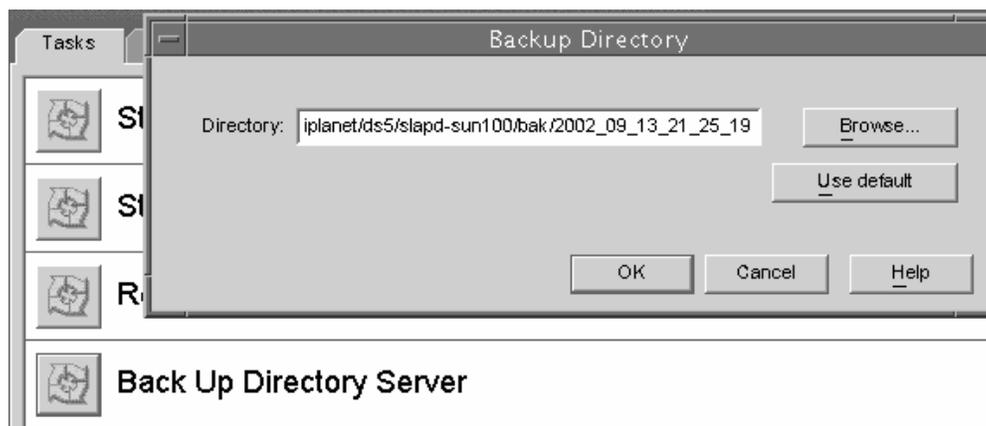


Figure 30.22 Backup Directory Window

As shown in Figure 30.22, the Backup Directory window will appear. This window lets you save the LDAP directory's information in a backup directory whose name includes the date and time. The database contents and index files are included in the backup directory. The directory's name is in the format of `YYYY_MM_DD_HH_MM_SS`.

12. Left click on the Use default button a couple of times (wait a few seconds between clicks).
Notice that the name of the directory changes with the new time.
13. Left click on the OK button.
14. After the backup is complete, left click on the Status Logs... button
15. Left click on Close to close the "Status For Backup" dialog.
16. Left click on Close to close the "Backup" dialog.
17. Left click on the Restore Directory Server icon. The Restore Directory window will appear, as shown in Figure 30.23.

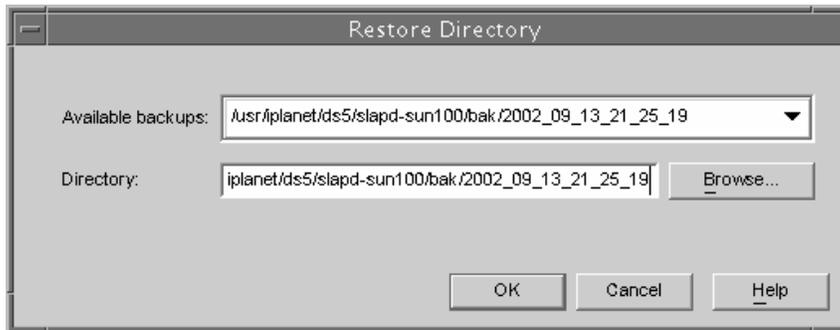


Figure 30.23 Restore Directory Window

18. In the Available backups field, click on the down arrow and select the latest backup.
19. In the Directory field, retain the default (this should be similar to the Available Backups field).
20. Left click on the OK button.
A Confirmation Needed screen pops up, that informs the administrator that the LDAP server will need to go offline.
21. Left click on the Yes button on the Confirmation Needed dialog.
22. Left click Close when the restore is complete.
23. After the restore is finished, left click on the Start Directory Server button.
*There should be a message that says **The server sun100 is already running**. This is a quick way to ensure that the server really did start again.*
24. Left click on the Export Databases icon. The Export Databases window appears.
25. In the "LDIF file" text box, type the name **myexport.ldif**
*The extension is dot (.) **L D I F** (in lower case).*

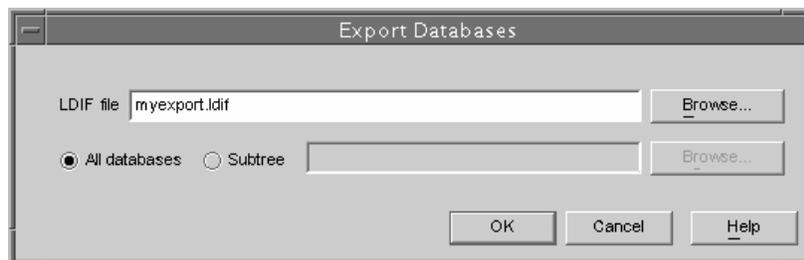


Figure 30.24 Export Database Window

Note: You can left click the "Browse..." button to select a directory to back up the file into. The Subtree radio button can be used to save only part of the LDAP tree instead of the whole tree.

26. Left click on the OK button.
You will see a warning message about having access rights to the file. The server should have the proper access, so the file should be created without any problems.

If for some reason there is an access issue, create a file with the following commands:
touch myexport.ldif
chmod 777 myexport.ldif
27. Left click on Status Logs...
Some versions of the LDAP Directory Server do not have this button in the LDAP homepage, so you may not be able to perform this step.
28. Check for any error messages. If you see the message **Export finished**, the export was completed.

29. Left click on Close to close the Status for Export dialog box.
 30. Left click on Close to close the Status Logs dialog box.
 31. Minimize all the console windows by clicking on the minimize button on the top left of the screen. It looks like a dot in a square.
 32. Open a new Terminal window.
 33. Type the command `find /usr -name myexport.ldif`
 34. Type the command `cd <directory-with-the-myexport.ldif-file>`
For example, you might type `cd /usr/iplanet/ds5/slapd-sun100/ldif/`
 35. Type the command `more myexport.ldif`
 36. Examine the contents of the `myexport.ldif` file. This file saves the contents of the DIT tree in a readable text format.
 37. Type the command `more Siroe.ldif`
Read through this LDIF file. Sun did a great job of creating an example LDIF directory for a fictitious company.
 38. To restore the Directory Server Console windows, close the Terminal window and double-click on the iPlanet Console and sun100.aassdfff.com icons.
39. Left click the Import Databases icon.

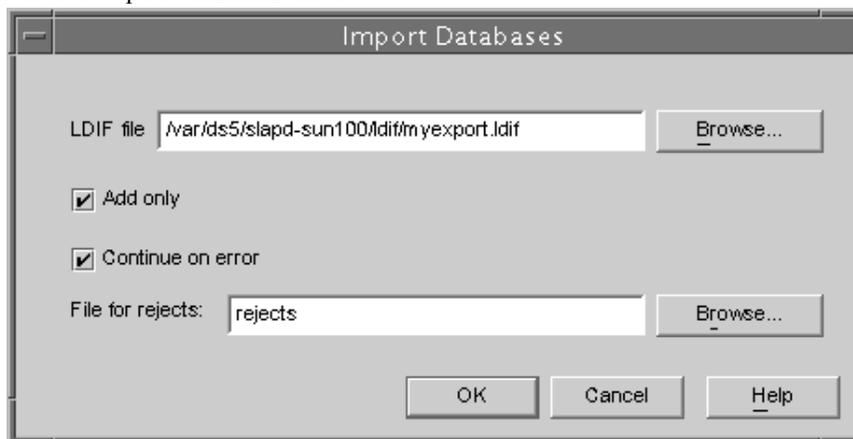


Figure 30.25 Import Databases Window

40. In the LDIF file input box, type the name of the directory that contains your exported LDIF file.
41. Left click on the top Browse... button
42. Highlight the `myexport.ldif` file.
43. Left click on Open.
44. Left click on Cancel (*don't import the database; it's very time consuming*).

Note: By default, exporting the entire directory as an LDIF does not export some of the system configuration data. You can use the Backup and Restore tools with The Directory Server to save more complete system-wide information.

45. Left click on Console in the Menu bar.
46. Left click on Exit

Lesson 30.7 More LDAP Administration

In this lesson readers will create a new group named Accounting Managers. Inside this group they will add an extra object called *businessCategory* to the Accounting Managers group. The group will be populated with the Directory Administrators group and the admin account.

This lesson assumes that the Directory and Administration servers are running.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command `/usr/sbin/directoryserver startconsole &`
4. Log in as the User ID: **admin** with your password.
5. In the left View pane, click on the icon next to the aassdff.com administrative domain.

This could be a different domain depending on the setup of your test system.

6. Left click on the Turner icon next to Server Group.
7. Left click on the Directory Server (sun100) icon.
8. Left click on the **O**pen button located on the top right corner of the right-hand screen.
9. Left click on the Directory tab.
10. Left click on the aassdff directory in the left-hand View pane.
11. Left click on **O**bject in the menu bar.
12. Left click on **N**ew ▶
13. Left click on **G**roup...

The Create New Group window will appear, as shown in Figure 30.26.

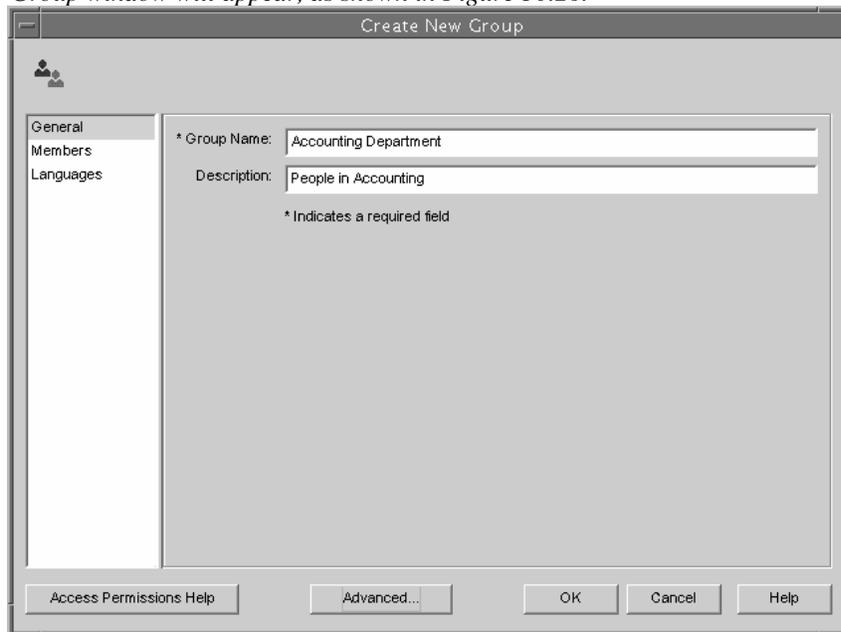


Figure 30.26 Create New Group Window

14. Fill in the fields as follows:
 - Group Name: Accounting Department
 - Description: People in Accounting
15. Left click on the Advanced... button
This button is located at the bottom of the window.
16. Left click on the Add Attribute button (at the lower right in Figure 30.27). You'll see the Add Attribute window, as shown in Figure 30.28.

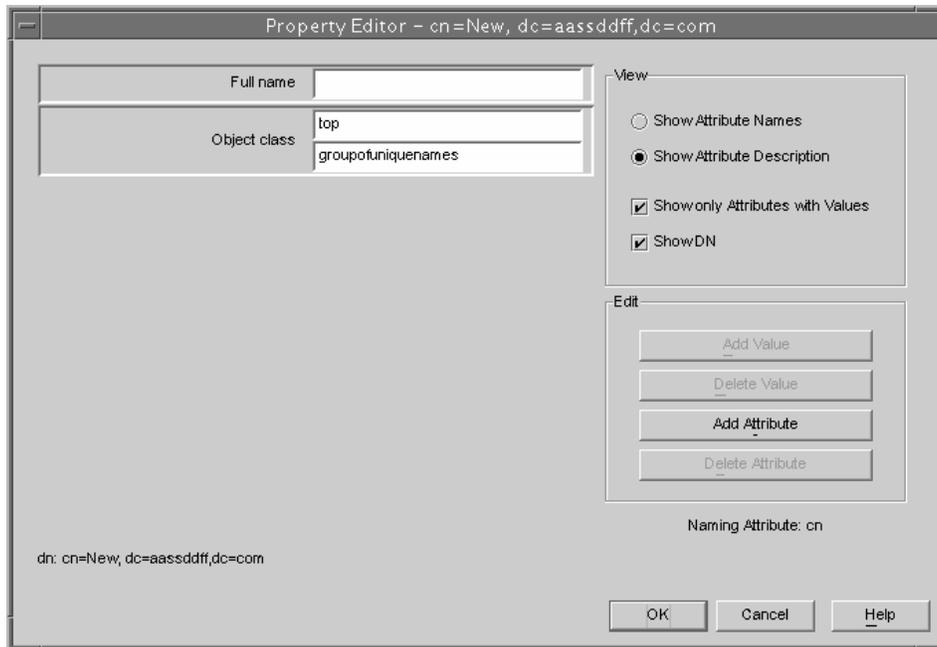


Figure 30.27 Property Editor Window

You will see the Add Attribute window, as shown in Figure 30.28.

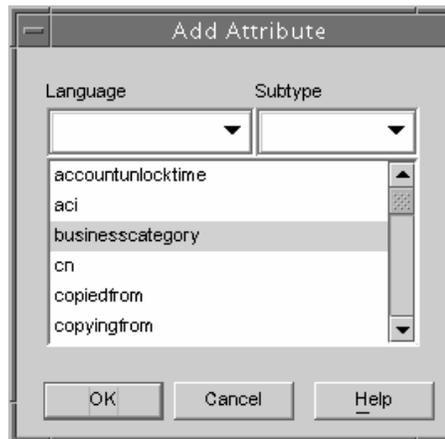


Figure 30.28 Add Attribute Window

17. Left click on businesscategory.
You will see the Property Editor window, as shown in Figure 30.29 (reduced view):

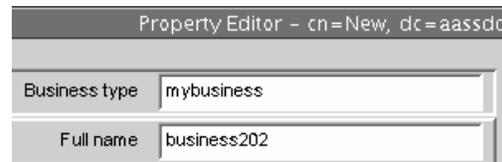


Figure 30.29 Property Editor Window (Reduced View)

18. Left click on OK to close the Add Attribute window.
You will see the Property Editor (Figure 30.27). Left click on the radio buttons next to “Show Attribute Names” and “Show Attribute Descriptions” to see the object’s name and description change.
19. Type **mybusiness** in the "Business type" text box.
20. Type **business202** in the "Full name" text box.
21. Left click on OK to close the Property Editor window.
22. In the Create New Group window, left click on Members in the left pane.
23. Left click on the Static Group tab.
24. Left click on Add.
There should be a popup window with the title “Search users and groups,” as shown in Figure 30.30.

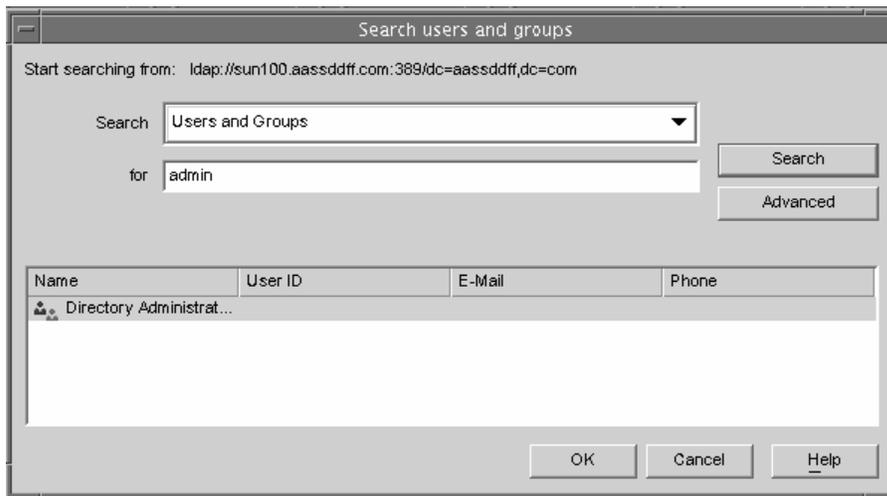


Figure 30.30 Search Users and Groups Window

25. Type **admin** in the "for" text box and click on Search.
26. Left click on OK.

Lesson 30.8 Create Another LDAP Administrator

In this lesson readers will create another Administrator for the LDAP server. This user is Adam Smith, with the login id of `asmith`. The new user has administration rights to the `aassdff.com` domain.

This lesson assumes that the directory and admin servers are running.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command `/usr/sbin/directoryserver startconsole &` to start the Console.
4. Log in as the User ID: `admin` with your password.



5. In the left View pane, click on the Turner icon next to your administrative domain.
6. Left click on the Turner icon next to the Server Group.
7. Left click on the Directory Server (sun100) icon
8. Left click on the Open button located in the top left corner of the right-hand screen.
9. Left click on the Directory tab.

In the left View pane is a window that shows the DIT for the LDAP server.

10. Right click on the sun100.aassdfff.com:389 icon at the top of the tree.
11. Select Properties from the popup menu.

A window should appear with the title "Property Editor – This shows the properties of the LDAP server."



12. Left click on the Cancel button.
Be sure not to modify any of the LDAP server's properties.
13. Left click on the aassdfff directory.
14. Right click on the aassdfff directory, but do not release the right mouse button.
15. Select New and then select User...

The Create New User window will appear, as shown in Figure 30.31.

Figure 30.31 Create New User Window

16. Create a user with the following attributes

First Name:	Adam
Last Name:	Smith
Common Name(s):	Adam Smith
User ID:	asmith

29. Left click on Set Access Permissions...
30. Left click on Add...
31. In the "for" text window, type **asmith**
32. When asmith user account is created, click on OK.
33. In the Set Permission Dialog, click on OK.
34. Left click on Console.
35. Left click on Exit.

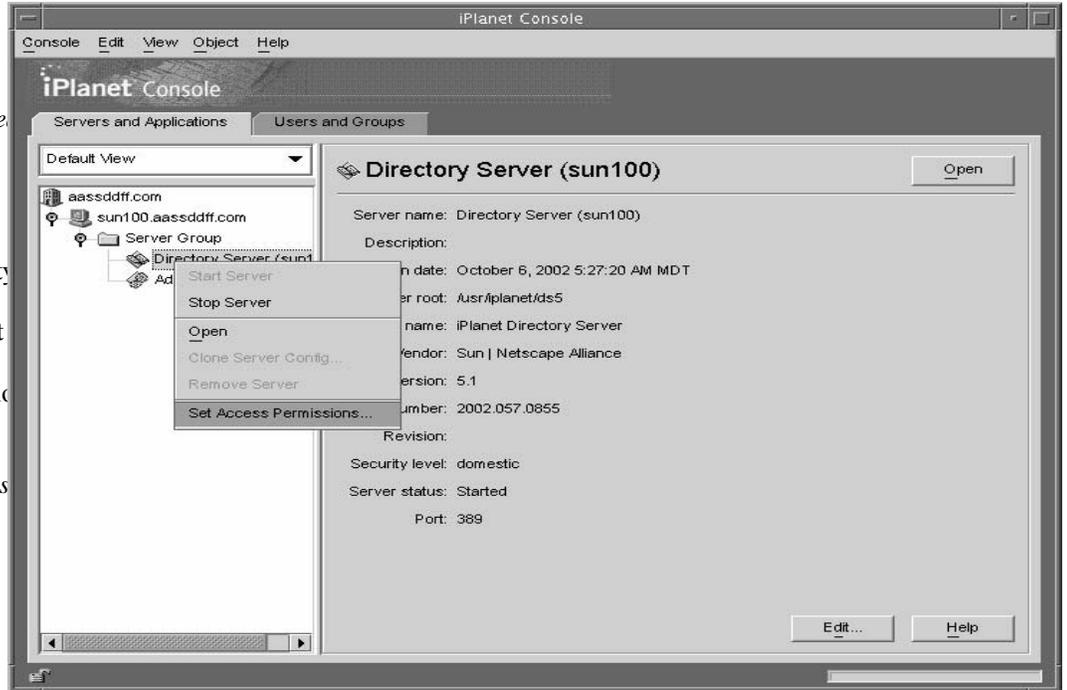


Figure 30.34 Directory Server in the Main iPlanet Console Window

Lesson 30.9 Monitoring the Directory Server's Performance

In this lesson readers will use the Status tab to monitor the LDAP server's performance.

This lesson assumes that the directory and admin servers are running.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command `/usr/sbin/directoryserver startconsole &` to start the Console
4. Log in as the User ID: admin with your password.
5. In the left View pane, click on icon next to the aassdff.com administrative domain.



- This could be a different domain, depending on the setup of your test system.*
6. Left click on the Turner icon next to the Server Group.
 7. Left click on the Directory Server (sun100) icon.
 8. Left click on the Open button located in the top left corner of the right-hand screen.
 9. Left click on the Status tab.
 10. Left click on the sun100.aassdff.com:389 icon.

You'll see a window like the one shown in Figure 30.35.

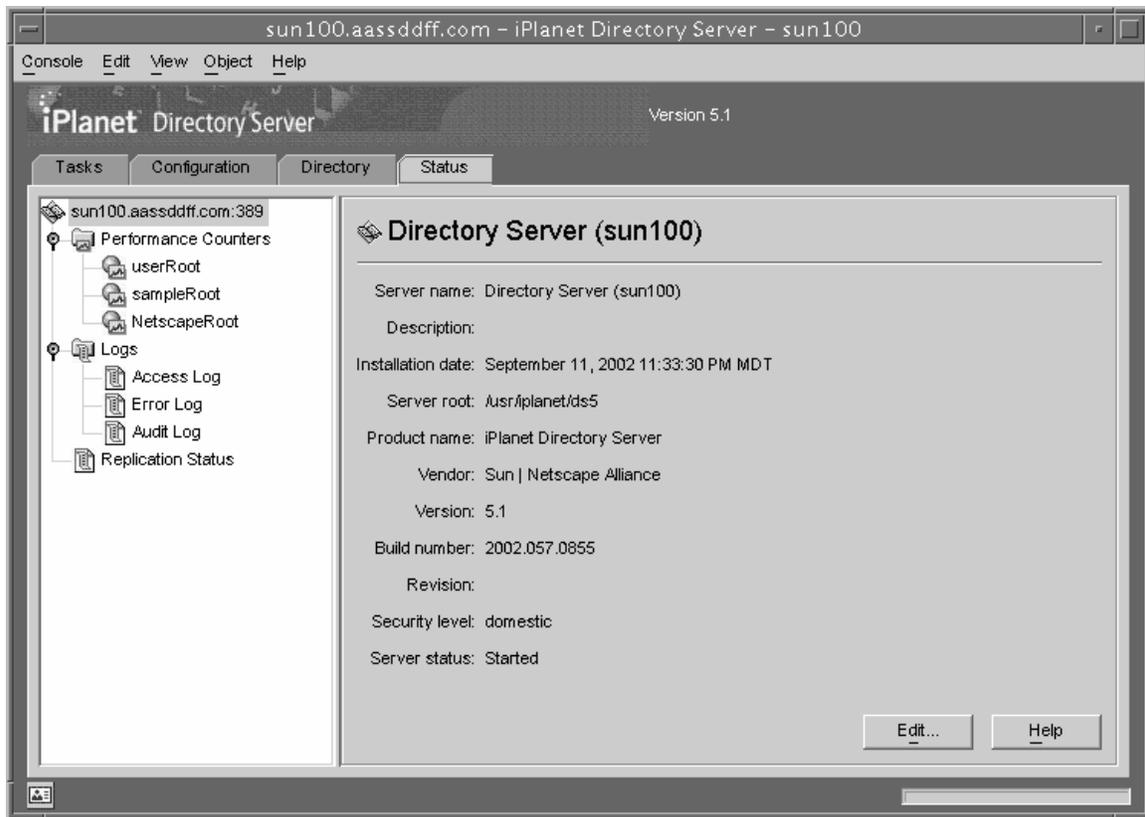


Figure 30.35 The sun100.aassdfff.com Entry in the Status Tab

This gives the same generic information seen on the initial Directory Server information screen that was used in the initial Console display.

11. Left click on Performance Counters.

Explore the Performance Counters in detail, then exit the Console

Commands in the /usr/iplanet/ds5/slaped-`<hostname>` Directory

Although many LDAP management functions are best performed from a GUI window, it's a good idea to know how to do certain operations from the command line. The following commands are located in the `/usr/iplanet/ds5/slaped-<hostname>` directory.

- db2bak** Creates a backup of the Directory Server.
- bak2db** Restores the Directory Server from a backup.
- db2ldif** Exports the database to an LDIF file. An LDIF file is a plain text file that can be easily viewed and edited with any text program such as **vi**, **cat** or **more**.
- ldif2db** Imports an LDIF file into the database.
- monitor** Used to monitor the LDAP server.

Lesson 30.10 Using Command Line Utilities

In this lesson readers will work with various command line utilities to back up the Directory Server. This lesson will also introduce readers to the **monitor** command line utility.

In this lesson readers will also configure the LDAP server for a roaming user named *asmith*. In the next lesson, the roaming user's Netscape browser will use the Directory Server to store Roaming Profile information. A roaming user is a user who logs in from more than one location. This is typically a sales person or someone else who may even log in from a remote location. In the lesson a roaming user's Netscape browser settings (such as bookmarks and fonts) will follow the user wherever the user logs in.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command **PATH=\$PATH:/usr/iplanet/ds5/slaped-sun100**
4. Type the command **export PATH**
This temporarily adds the /usr/iplanet/ds5/slaped-sun100 directory.
5. Type the command **date**
This displays the system's date and time
6. Type the command **db2bak**
The db2bak command is used to back up the database from the command line. This command saves the same files and directories as the Backup the Directory Server icon in the GUI.
7. Type the command **cd /usr/iplanet/ds5/slaped-sun100/bak**
8. Type the command **ls**
The ls command shows all the backup jobs performed for the Directory Server.
9. Type the command **/usr/iplanet/ds5/slaped-sun100/saveconfig**
This Bourne shell script saves the configuration of the LDAP server. This command creates an LDIF file in the /usr/iplanet/ds5/slaped-sun100/confbak directory. This file's name is based on the system's date and time.
10. Type the command **cd /usr/iplanet/ds5/slaped-sun100/confbak**
Now let's look inside the /usr/iplanet/ds5/slaped-sun100/confbak directory where the backup file just created was saved.
11. Type the command **ls**
There should be an LDIF file for the LDAP server, with a name like 2002_07_18_135423.ldif
12. Type the command **more 2002_07_18_135423.ldif**
This shows the contents of the LDIF file. Use whatever filename is appropriate for your setup.
13. Type the command **/usr/iplanet/ds5/slaped-sun100/monitor**
This shows the LDAP server's current configuration.
14. Type the command **cd /usr/iplanet/ds5/slaped-sun100/ldif**
15. Create a file called **roaming.ldif** in the **ldif** directory with the following contents:

```
dn: ou=Roaming,dc=aassdff,dc=com
objectclass: top
objectclass: organizationalUnit
ou: Roaming
```

This file creates a place in the directory tree that will contain roaming profiles.

16. Type the following command. Put everything on one line, even though it is shown in two lines here.
`/usr/iplanet/ds5/slaped-sun100/ldif2ldap "admin" admin123 /usr/iplanet/ds5/slaped-sun100/ldif/roaming.ldif`
This will import the LDIF file that you just created into the directory. If you have any problems, check the username and password used on the command line, and check the contents of your LDIF file.
17. Create a file called **profile-asmith.ldif** in the **ldif** directory with the following contents:

```
dn: nsLIProfileName=asmith,ou=Roaming,dc=aassdff,dc=com
```

```
changetype: add
objectclass: top
objectclass: nsLIProfile
nsLIProfileName: asmith
owner: uid=asmith,dc=aassdfff,dc=com
```

*This creates a roaming profile for the **asmith** user. You should see a response like adding new entry nsLIProfileName=asmith,ou=Roaming,dc=aassdfff,dc=com*

18. Type the following command. Put everything on one line, even though it is shown in two lines here.

```
/usr/iplanet/ds5/slapd-sun100/ldif2ldap "admin" admin123
/usr/iplanet/ds5/slapd-sun100/ldif/profile-asmith.ldif
```

This will import the LDIF file that you just created into the directory. If you have any problems, check the username and password used on the command line, and check the contents of your LDIF file. You should see a response like

adding new entry ou=Roaming,dc=aassdfff,dc=com

For more information on LDIF files, see RFC 2849 - The LDAP Data Interchange Format (LDIF) - Technical Specification. This is available at <http://www.ietf.org/rfc/rfc2849.txt>.

Lesson 30.11 Using the Netscape Browser with LDAP

In this lesson readers will configure the Netscape browser to use the Directory Server to store Roaming Profile information. A roaming user is a user who logs in from more than one location. This is typically a sales person or someone else who may even log in from a remote location. In this lesson, a roaming user's Netscape browser settings (such as bookmarks and fonts) will be set up to follow the user wherever the user logs in.

This lesson assumes that you have created the asmith user and have imported the LDIF files from the earlier lessons.

1. Log in as the root user.
2. Open a Terminal window.
3. Left click the Netscape Web browser icon.
The Netscape Web browser icon looks like a globe with red clock hands on the surface.
4. Left click on the Edit menu choice.
5. Left click on Preferences.
6. A window will appear with the title "Netscape: Preferences."
7. In the left pane, left click on the ▾ Roaming User menu item.
The window shown in Figure 30.36 will appear.



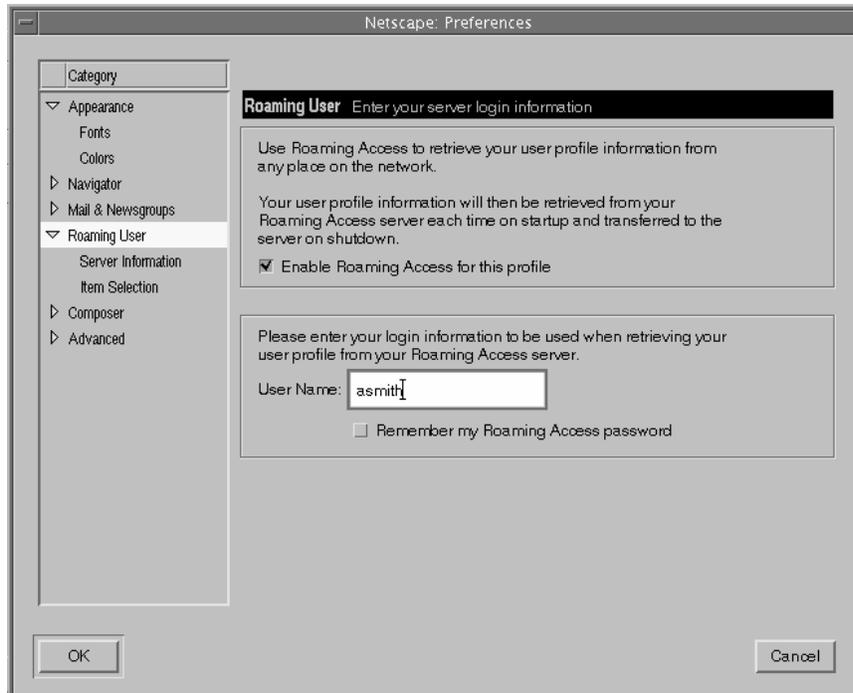


Figure 30.36 The Roaming User Preferences Window

8. Left click the check box entitled "Enable Roaming Access for this profile."
9. In the right pane, in the "User Name" text box, type **asmith**
10. In the left pane, select Server Information.

The window shown in Figure 30.37 will appear.

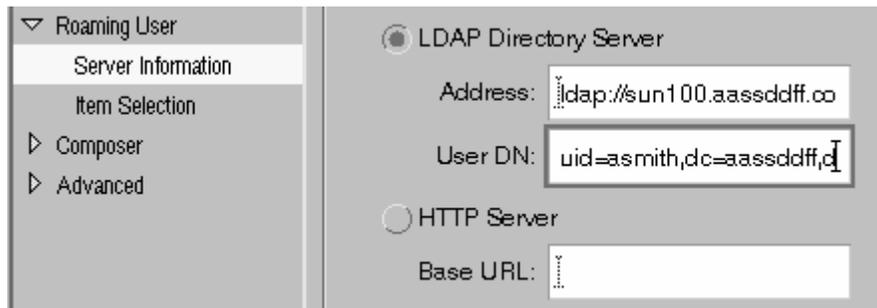


Figure 30.37 Roaming User Server Information Window

11. In the "Address:" text box, type (all on one line):
ldap://sun100.aassddff.com/ndLIProfileName=asmith,ou=roaming,dc=aassddff,dc=com
12. In the "User DN:" text box, type in **uid=asmith,dc=aassddff,dc=com**
The user asmith was created in the previous lesson.
13. Left click on the OK button. *You will see a window that tells you that your changes will take effect the next time you restart Communicator.*
14. Left click on the OK button.
15. Close Communicator by selecting File in the Menu bar and left clicking on Exit.

16. Confirm that you want to exit Netscape by left clicking on the OK button.
17. Restart Netscape with the Globe icon.
18. You will be prompted for your login information. Enter the password for asmith (**asmith123**).
19. Your profile is now stored in the Directory and can be accessed from any configured client

LDAP Run Control Scripts

The installation of the Directory Server on Solaris 9 creates a Bourne shell script called `/etc/init.d/directory` that starts and stops the Directory Server and the Administration Server.

The following files are symbolically linked) to the `/etc/init.d/directory` file. (A symbolically linked file is an empty file that points to another file; you can use the `ls -l` command to view the symbolic link.)

```
/etc/rc0.d/K41directory
/etc/rc1.d/K41directory
/etc/rc2.d/S72directory
```

When the Solaris 9 server shuts down, the `K41directory` files gracefully shut down the Administration Server and the LDAP server. When the Solaris 9 server starts, the `S72directory` file gracefully starts the Administration Server and the LDAP server.

You might actually consider disabling the `/etc/rc2.d/S72directory` and only start the LDAP server when needed by hand. This is because when the LDAP server is running it could slow down a workstation to an unacceptable speed. To start the LDAP server by hand, type the command `/etc/init.d/directory start`.

WARNING

Never disable the LDAP server shutdown script `/etc/rc1.d/K41directory`.

If Solaris 9 shuts down and the LDAP server is not gracefully shut down in the process, the LDAP data could become damaged.

The following two scripts are set up when Solaris 9 is installed:

```
/etc/rc0.d/K41ldap.client
/etc/rc2.d/S71ldap.client
```

These scripts only prepare Solaris 9 to be an LDAP client. They are not involved with the LDAP server.

Lesson 30.12 Set Up Alternate Run Control Scripts (optional)

This lesson shows how to create a custom run control script that automatically starts and shuts down the Administration Server and the LDAP server when Solaris 9 goes up and down. Remember, the installation of the LDAP server has already created start and stop scripts for that server. This lesson just shows how to make a custom LDAP start/stop script. This lesson is also a good for learning how to create an `rc` script, and to learn how the startup and shutdown process work.

This lesson is optional!

The script created in this lesson is an alternate rc script that is used to demonstrate some of the techniques used to create startup and shutdown scripts. Understand that the Directory Server and the Administration Server take up a lot of system resources.

Modify the following startup script to match the path where the **start-slaped** command and the **start-admin** command reside. This script basically starts the Directory Server with the **start-slaped** command. It also starts the Administration Server with the **start-admin** command. Name the script so that it is one of the last **/etc/rc3.d** start scripts to be executed. The script's name must start with a capital **S**, then two numbers and then a text name. To disable this script in the future, change the capital **S** to a lower case **s** so that the script will not be executed.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command **cd /etc/init.d**
4. Type the command **vi /etc/init.d/myldapscript**
In this file, put the lines shown below.

You will need to customize the values of the LDAP_PORT and ADMIN_PORT variables for your installation. You can also adjust the SLEEPTIME variable to slow the process down for testing purposes. The default of 0 disables the pauses.

```
#!/sbin/sh
#
# directory: start and stop LDAP directory services
#
PATH=/bin:/usr/local/bin:/usr/bin:/bin
GREP=/usr/bin/grep
NETSTAT=/bin/netstat
PGREP=/usr/bin/pgrep
LDAP_PORT=389
#
# modify ADMIN_PORT to match your LDAP server's administration port number
#
ADMIN_PORT=25791
SLEEPTIME=0
case $1 in
'start')
    echo "Starting slapd ... \c"
    (/usr/sbin/directoryserver start && echo "Done.") \
    || echo "failed: error $?"
    echo Waiting ...
    sleep ${SLEEPTIME}

    echo "Starting directory admin interface ... \c"
    (/usr/sbin/directoryserver start-admin && echo "Done.") \
    || echo "failed: error $?"
    ;;
'stop')
    echo "Stopping directory admin interface ... \c"
    (/usr/sbin/directoryserver stop-admin && echo "Done.") \
    || echo "failed: error $?"
    echo Waiting ...
    sleep ${SLEEPTIME}
```

```

    echo "Stopping slapd ... \c"
    (/usr/sbin/directoryserver stop && echo "Done.") \
    || echo "failed: error $?"
;;

'restart')
    $0 stop
    echo Waiting ...
    sleep ${SLEEPTIME}
    $0 start
;;

'status')
    echo --- Looking for iPlanet-related processes ...
    ${PGREP} -fl iplanet
    echo --- Looking for iPlanet-related ports in use ...
    ${NETSTAT} -an | egrep "${ADMIN_PORT}|${LDAP_PORT}|ldap"
;;

*)
    echo -e "\nusage: $0 {start|stop|restart|status}\n"
;;

    esac
    exit 0
# end of script

```

Note that it is important that the Administration Server start before the Directory Server, and that the Administration Server stop before the Directory server. The Administration Server relies on the Directory Server for proper startup.

- Change the permissions and ownership of the resulting `myldapscript` file with the following commands:

```

chown root:sys /etc/init.d/myldapscript
chmod 744 /etc/init.d/myldapscript

```

These commands make the ownership and permissions of the rc script like most of the others in the `/etc/init.d` directory.

- Enter the `ls -la` command to see the results of these permission changes.

```

ls -la /etc/init.d/myldapscript
-rwxr--r-- 1 root  sys   1353 Sep 17 19:44 /etc/init.d/myldapscript

```
- Test the script by executing it with the non-invasive `status` parameter:

```

/etc/init.d/myldapscript status

```

You should see a display like:

```

--- Looking for iPlanet-related processes ...
375 ./ns-slapd -D /usr/iplanet/ds5/slapd-sun100 -i /usr/iplanet/ds5/slapd-sun100/lo
401 ./uxwdog -d /usr/iplanet/ds5/admin-serv/config
402 ns-httpd -d /usr/iplanet/ds5/admin-serv/config
404 ns-httpd -d /usr/iplanet/ds5/admin-serv/config
--- Looking for iPlanet-related ports in use ...
*.389      *.*      0    0 49152    0 LISTEN

```

```
192.168.254.70.32786 192.168.254.70.389 49152 0 49152 0 ESTABLISHED
192.168.254.70.389 192.168.254.70.32786 49152 0 49152 0 ESTABLISHED
```

Review the display for unexpected error messages. If you get error messages, go back to step 4 and check for typos and other errors in the script. Double check the directories and files that are referenced by the script to make sure that they exist and have the proper read/write/execute permissions.

Writing a “status” parameter into an rc script (in the section of the script that starts with the label 'status') is sometimes handy for quickly checking up on the status of a process. Review the script for typos if you get unexpected script errors.

8. Test the script more rigorously by passing it the **stop** parameter, then the **status** parameter, and then the **start** parameter, as shown below:

- a. Type the command:
`/etc/init.d/myldapscript stop`

You will see the output:

```
Stopping directory admin interface ... /usr/iplanet/ds5/stop-admin
Done.
Waiting ...
Stopping slapd ... /usr/iplanet/ds5/slapd-sun100/stop-slapd
Done.
```

- b. Type the command:
`/etc/init.d/myldapscript status`

You will see the output:

```
--- Looking for iPlanet-related processes ...
--- Looking for iPlanet-related ports in use ...
```

- c. Type the command:
`/etc/init.d/myldapscript start`

You will see the output:

```
Starting slapd ... /usr/iplanet/ds5/slapd-sun100/start-slapd
Done.
Waiting ...
Starting directory admin interface ... /usr/sbin/directoryserver start-admin
iPlanet-WebServer-Enterprise/6.0SP1 B11/09/2001 13:59
Info: Cache expiration set to 600 seconds
warning: daemon is running as super-user
[LS ls1] http://sun100.aassdff.com, port 25791 ready to accept requests
startup: server started successfully
Done.
```

These commands put the script through its “paces.” It is always a good idea to test an rc script before a server is restarted. There is nothing worse than a server that has been running for six months, gets rebooted, and then hangs on a bad startup/shutdown script. . . and no one knows why.

9. Type the command `cd /etc/rc2.d`
10. Type the command `ln -s /etc/init.d/myldapscript S/etc/rc2.d/S72myldapscript`
This creates a link from the startup script to the /etc/rc2.d/S72myldapscript script.
11. Type the command `ls -ls /etc/rc2.d/72myldapscript`

You should see the following output:

- ```
lrwxrwxrwx 1 root other 21 Sep 17 20:12 S72myldapscrip -> ../init.d/myldapscrip
```
12. Type the command `ln -s /etc/init.d/myldapscrip /etc/rc0.d/K41myldapscrip`  
*Scripts that begin with K in the /etc/rc\*.d directories are automatically passed the kill parameter when the system is shutting down.*
  13. Type the command `ls -la /etc/rc0.d/K41myldapscrip`  
*This command should produce the output:*

```
lrwxrwxrwx 1 root other 21 Sep 17 20:12 K41myldapscrip -> ../init.d/myldapscrip
```
  14. Reboot the system to make sure that the shutdown and startup scripts are working properly.  
*For more information on how startup and shutdown scripts work, read the /etc/init.d/README file, and see Chapter 12 ("Bourne Shell Scripts") in this book.*

## Lesson 30.13 Set Up a Client Profile

In this lesson readers will create a client profile. A client profile is needed to set up LDAP clients. In this lesson the `ldapclient` command will be used to set up a profile named `goodprof.ldif`

1. Log in as the root user.
2. Open a Terminal window.
3. Type the following rather lengthy command (all on one line):  
*Note that your IP address may be different from the one shown here.*

```
ldapclient genprofile -a profileName=goodprof -a
defaultSearchBase=dc=aassddff,dc=com -a "defaultServerList=192.168.0.4" >
goodprof.ldif
```

*The profileName can be the whatever you desire. It is usually a name that is easy to remember, such as engineeringprofile or salesprofile. LDAP clients use these profiles during their configuration*

4. Type the command `more goodprof.ldif`  
*This file can be placed on another client, so that the client knows how to locate its default Directory Server.*

For more information about setting up LDAP clients, see Sun Blueprints or <http://docs.sun.com>, or for troubleshooting, see <http://sunsolve.sun.com>.

### The LDAP Client

Unfortunately, the root master LDAP server cannot also be its own LDAP client, in the true sense of an LDAP client. So, because it would be impossible to make the LDAP server be its own LDAP client, this chapter does not have a lesson on LDAP clients. However, this section does explain some of the major principles of setting up Solaris 9 as an LDAP client.

- LDAP clients must use the keyword `ldap` in the `/etc/nsswitch.conf` file. The LDAP client uses the `ldap_cachemgr` daemon to access information like hostnames, passwords, and groups from the LDAP server.
- The text file `/var/ldap/ldap_client_file` keeps some client profile information, such as:

|                                  |                                               |
|----------------------------------|-----------------------------------------------|
| <code>bindTimeLimit</code>       | The time-out value for a TCP/IP connection    |
| <code>defaultServerList</code>   | List of LDAP servers                          |
| <code>preferredServerList</code> | Which servers to contact first                |
| <code>profileTTL</code>          | Time the client should download a new profile |
| <code>searchTimeLimit</code>     | The time-out value for an LDAP search         |

- The text file `/var/ldap/ldap_client_cread` keeps the **ProxyDN** (Proxy Distinguished Name) and **proxyPassword** variables. The **proxyPassword** is encrypted. Only the root user can access this file. Even if the file is compromised by a hacker, the password is encrypted.
- The `ldap_cachemgr` is started from the `/etc/rc2.d/s71ldap.client` script.
- Solaris 9 LDAP servers support what are known as *client profiles*. These profiles store all the configuration information needed by a client. The profile is created by the `ldapclient` command with the `genprofile` argument.
- The easiest way to make a Solaris 9 workstation or server into an LDAP client is to use the `ldapclient` utility. This utility initializes the LDAP client machine, based on a profile on the LDAP server. The `ldapclient` utility sets up the proper files, directories and configuration information in local cache files.
- To view the LDAP information sent to a client, type the command `ldapclient list`

This will show the LDAP information that the LDAP server sends to the client. For readability, the output is in text form, rather than the actual binary format.

- To modify a client's attributes in the configuration files, type the command `ldapclient mod -a <attributes_to_modify>`
- The command `ldapclient uninit` can be used to return an LDAP client to its original condition. To manually remove LDAP from a client, give the following commands:

```
rm /etc/nsswitch.conf
rm /var/ldap/ldap_client_file
rm /var/ldap/ldap_client_cread
rm /etc/nsswitch.conf
cp nsswitch.files nsswitch.conf
cp /etc/rc2.d/s71ldap.client /etc/rc2.d/s71ldap
/etc/init.d/ldap.client stop
```

## Other Client-Related LDAP Information

This section discusses some LDAP terms and concepts that the reader should know when reading LDAP material.

### LDAP Client Access

LDAP clients can connect to an LDAP server in one of three ways.

**Anonymous Connection** Anonymous connections only let a client connect to public sections of the LDAP directory. When the client connects, only read access is permitted. A competent LDAP administrator would never grant write access for an anonymous connection, because a hacker could damage the DIT.

**Proxy Connection** With a proxy connection, the client is required to authenticate with the LDAP server. Not all clients have the same proxy access. Different proxy agents can be created on the LDAP server. The client is required to use an encrypted password to connect to the LDAP server. Once the authentication is approved, the client binds to the directory using a proxy account. The client in most cases can read and write to the LDAP DIT.

## Proxy Anonymous

Proxy anonymous can be thought of as a hybrid between a proxy connection and an anonymous connection. If a client can authenticate as a proxy client, the client is granted proxy access to the DIT. If the client cannot authenticate as a proxy client (an expired password is the most common reason), the client is granted anonymous access to the system.

The client's authentication password is stored in the **proxyDN** and **ProxyPassword** variables in the `/var/ldap/ldap_client_cred` file. Only the root user can access this file. As an added measure of security, the variables **ProxyPassword** and **proxyDN** are encrypted.

## Understanding the ldapclient Utility

The **ldapclient** utility has three main purposes. It can be used to set up an LDAP client machine, to list the contents of the LDAP client cache in plain text and to restore network services for LDAP clients. This utility can also be run on the command line to initialize an LDAP client.

The **ldapclient** command line utility supports the following options

**ldapclient** <options>

- a** Name of an attribute and its value. This looks something like  
credentialLevel=proxy  
proxyDN=cn-proxyagent,ou=myprofile,dc=aassddff,dc=com
- q** Quiet mode
- v** Verbose mode

If an attribute is not specified, default values will be set by the **ldapclient** utility. Any files that have been changed during the LDAP setup are in the `/var/ldap/restore` directory.

For the LDAP client to be set up, a client LDAP profile must be set up on the LDAP server. This is beyond the scope of this book. There are many ways to set up an LDAP profile on an LDAP server.

**ldapclient init** <IPaddress-of-LDAP-server>

This command sets up the client using the default profile on the LDAP server.

## The ldap\_cachemgr Daemon

The **ldap\_cachemgr** daemon keeps a list of active LDAP servers. This daemon must run as root. The **ldap\_cachemgr** utility updates LDAP client configuration when the LDAP cache expires.

## LDAP Security

The client's credentials are not stored in the client's profile. In Solaris 9, the client's proxy credentials are stored locally. TLS can be used for authentication. If for some reason TLS is not supported, Simple Authentication and Security Layer (SASL) can be used for secure password exchanges without TLS.

Two types of security can be enabled. When the client authenticates with the LDAP server, the password can be sent in clear text, or it can be encrypted. After the client authenticates with the server, the LDAP session can either be conducted in clear text or, it can be encrypted. If the LDAP session is not encrypted, a hacker might not be

able to gain access to a password, but could still gain valuable system information from the LDAP session itself. When choosing a security model, consider both of these types of interactions with the LDAP server.

The client can authenticate with the server using the following methods:

### Low-security Authentication Methods

- none** The client has no authentication to the LDAP server. This is basically an anonymous, unencrypted connection.
- simple** This the worst form of authentication. It sends a clear text password to the LDAP server. Only isolated test labs should use this. Any workstation using the **snoop** command could pick up LDAP passwords off the wire. If a server is being set up for the first time (without confidential information), the simple authentication can be used, just to make sure that authentication can occur. More advanced authentication models can then be used. Once the more secure authentication is enabled, simple authentication should be permanently disabled.
- sasl/cram-MD5** The client's authentication password is encrypted, but the LDAP session is not encrypted. This security model is better than simple, is still not the best one. If a workstation used the **snoop** command, the LDAP session and its information could be intercepted. Any information can be the first foothold into cracking a system.
- sasl/digest-MD5** A slightly more secure method than sasl/cram-MD5. As with sasl/cram-MD5, the client's authentication password is encrypted, but the rest of the LDAP session is not encrypted.

### Higher Security Authentication Methods

- tls:simple** The client authenticates using the simple method, but the LDAP session is encrypted. Unfortunately, this means that an intruder can then authenticate with a snooped password.
- tls:sasl digest-MD5** The client authenticates to the LDAP server using sasl/digest-MD5. The session is also encrypted, providing for full security during the LDAP session.
- tls:sasl cram-MD5** The client authenticates to the LDAP server using sasl/cram-MD5 encryption. The session is encrypted.

To use TLS, TCP ports 389 and 636 must be opened on the server, and on any firewall connections between the server and the clients. Note also that under the Directory Server 5.1, you cannot use TLS unless you are using the default ports.

## Understanding the `ldapaddent` Utility

The `ldapaddent` utility is used to add entries from the `/etc` directory to the LDAP server. This tool is primarily used when a NIS or NIS+ server is being replaced with an LDAP server. This utility is run on the client to publish its `/etc` information to the LDAP server. This utility can only be used if the `ldapclient` utility has been successfully executed.

## Lesson 30.14 Uninstall the Directory Server

Uninstalling the Directory Server is very easy. When you are ready to do so, follow the directions below.

### WARNING

This lesson uses the **rm -rf** command to remove the **/usr/iplanet** directory and all its subdirectories. *Be very careful this command. It deletes every file and directory specified after **rm -rf** and can easily destroy a server!*

1. Log in as the root user.
2. Type the command **/usr/iplanet/ds5/uninstall**  
You will see the following screen:

#### Sun-Netscape Alliance iPlanet Server Products Installation/Uninstallation

---

**The following are the iPlanet components currently installed on your machine:**

**Components with a number in () contain additional subcomponents which you can select using subsequent screens.**

1. iPlanet Directory Suite (2)
2. Administration Services (2)

**Select the components you wish to uninstall (default: all) [All]:**

3. Press Enter to select the [All] option.  
**Note:** *The uninstall process typically does not remove the **/usr/iplanet** directory and its subdirectories. The following steps remove those directories.*
4. Type the command **cd /usr**
5. Type the command **pwd**  
*Make sure the current working directory is **/usr**.*
6. Type the command **rm -rf iplanet**  
*As mentioned in the warning above, be very careful with the **rm -rf** command. Make absolutely sure that you have entered it correctly before you press the Enter key!*

### LDAP Command Line Utilities

This section describes various command line utilities that can be used with the LDAP server. To use these tools it is necessary to change to the directory where the tools are located. Alternatively, you can set the **LD\_LIBRARY\_PATH** environment variable.

Because of some rather obscure library path issues, these each of these commands is set up to be run from its directory. For now run each of these commands as follows:

1. Type **cd </path/to/the/command/>**
2. Type the command.

Tools in the **/usr/iplanet/ds5/shared/bin** directory

- ldapdelete**      Deletes entries in the LDAP server.
- ldapmodify**     Modifies the LDAP server's entries
- ldapsearch**     Connects to an LDAP server and then does an LDAP search based upon a filter.
- NativeToAscii**   Works with the **conv** directory. Converts between different character sets. It takes the SourceEncoding and displays output as TargetEncoding. For information on what conversions it offers, type the command **NativeToAscii -l** (small letter L).

Tools in the `/usr/iplanet/ds5/bin/slapd/server` directory:

- idsktune**        A good troubleshooting tool that shows errors and warning messages that might cause an LDAP server not to start.
- viewcore**       Can be used to view core files from an LDAP server crash.

### Important LDAP-related RFCs

For more information about LDAP, see the following RFCs:

- 1309      Technical Overview of Directory Services - Using the X.500 Protocol
- 1487      X.500 Lightweight Directory Access Protocol
- 1777      Lightweight Directory Access Protocol
- 2251      LDAPv3
- 2255      The LDAP URL Format
- 2307      An Approach for Using LDAP as a Network Information Service
- 2307bis   (A modified version of 2307) <http://www.padl.com/~lukeh/rfc2307bis.txt>
- 2798      Definition of the inetOrgPerson LDAP Object Class
- 2830      Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security
- 2849      The LDAP Data Interchange Format (LDIF) - Technical Specification

### References and Further Reading

The University of Michigan's LDAP server (historical information)  
<http://www.umich.edu/~dirsvcs/ldap/>

DirectoryMark - an LDAP benchmarking utility  
<http://www.mindcraft.com/directorymark/>

PADL Software – maker of various LDAP tools  
<http://www.padl.com/>

Directories and X.500: An Introduction  
<http://www.nlc-bnc.ca/9/1/p1-244-e.html>

An LDAP Roadmap and FAQ by Jeff Hodges, Kings Mountain Systems  
<http://www.kingsmountain.com/ldapRoadmap.shtml>

## **Key Points to Remember**

This chapter covered a lot of ground. The first part of this chapter deals with the installation of the LDAP server. This installation is rather straightforward. Follow Lesson 30.1 before you install the LDAP server. The LDAP server can be started and shut down with run control scripts or can be started or shut down by hand. Because it is very resource intensive, start the LDAP server only if it is needed.

## Chapter 31 DHCP

---

### Lessons in This Chapter

|                                                                                    |       |
|------------------------------------------------------------------------------------|-------|
| Lesson 31.1 Set up a Server as a DHCP Relay Agent .....                            | 31-6  |
| Lesson 31.2 Creating a DHCP Server with the DHCP Manager .....                     | 31-10 |
| Lesson 31.3 Creating a DHCP Client .....                                           | 31-23 |
| Lesson 31.4 Configuring Macros And Options .....                                   | 31-25 |
| Lesson 31.5 Exporting DHCP Server Information .....                                | 31-28 |
| Lesson 31.6 Exporting DHCP Server Information .....                                | 31-31 |
| Lesson 31.7 Using the <code>pnadm</code> Command.....                              | 31-35 |
| Lesson 31.8 Starting and Stopping the DHCP Server From the Command Line .....      | 31-36 |
| Lesson 31.9 Using <code>dhcpconfig</code> to Clear and Restore a DHCP Server ..... | 31-36 |
| Lesson 31.10 Using Command Line Utilities with a Solaris 9 DHCP Client.....        | 31-37 |
| Lesson 31.11 Working with the <code>dhtadm</code> Command .....                    | 31-37 |

### Introduction

One problem with a large network is that every system on a standard IPv4 network (such as network printers, Microsoft Windows 2000 workstations, Windows 98 workstations and Solaris workstations) needs to be configured with three key pieces of information to work on a network. Each system needs to have an IP address, a subnet mask and the IP address of the default router. If the network topology changes, these pieces of information need to be changed on all the systems affected by the change. Unfortunately it is a time consuming process to change all this information on each system. Every network device also requires that staff know how to safely change the network settings without accidentally disabling the network services on the device.

DHCP (Domain Handling Control Protocol) was invented to reduce the burden of configuring network devices. A DHCP server gives a client its IP address, subnet mask and default router. If the network's topology changes, the network engineer simply changes the DHCP server's information. The only configuration necessary on the client is to make the client refresh its DHCP information. Devices that have static IP address can live in a DHCP network segment, provided the IP address is not used and the DHCP server is properly configured.

For diskless clients, DHCP is an absolute necessity. When a diskless client starts, it sends out a BOOTP packet using the network broadcast address 255.255.255.255. Basically, this packet asks the question "This is my Ethernet address, what are my network settings?" on the network segment.

The DHCP server hears the request and then leases an IP address, subnet mask and default router to the diskless client. Other network information can also be given to the client through DHCP. Some routers can act as a BOOTP relay agent and direct the client's broadcast message to a DHCP server, even to servers not on the same network segment.

DHCP is also very useful for laptop users and roaming users. These users need different network settings, depending on where they attach their systems. This information can be provided by a DHCP server in the network segment. In most cases, not all the roaming users will have their computer turned on at the same time. Because of this use pattern, most networks can assign a small pool of IP addresses to serve a much larger pool of actual clients.

Solaris 9 has a built in DHCP server. This server is an industry compliant DHCP IPv4 server. However, this DHCP server does not work with IPv6. Understand that IPv6 takes care of a lot of its own housekeeping.

There are two main tools used to configure DHCP. The DHCP manager is a Java-based GUI tool that has various configuration wizards and view panes. The command line tool **dhcpconfig** can perform most of the same functions as the DHCP manager, except that it works without the need for a graphical display system or X server.

## Quick Tip

- Sometimes when readers set up a DHCP (Domain Handling Control Protocol) server they notice some DNS (Domain Name Service) server settings, such as the DNS server's IP address and domain. Understand that these settings only point to a DNS server. They do not mean that you are setting up a DNS server.
- If you type **ping <hostname>.<domain>** and you get the message **<hostname>.<domain> is alive**, this most likely indicates that a DNS server is active on this domain and is resolving this host and this domain.
- If you type **ping <hostname>.<domain>** and you get back the response: **<hostname>.<domain> not found**, this indicates that a DNS server is not actively resolving this workstation's DNS name.
- Understand that a DNS server and a DHCP server are different types of servers, and that DHCP settings only *point* to a DNS server. The DHCP server can not perform hostname/domain resolution; only a DNS server performs that role.

### How DHCP Works

DHCP generally works as follows:

1. A client that does not have an IP address starts up. It sends out what is known as a **BOOTP** packet. This packet has the client's Ethernet card address (example **0:3:ba:4:c1:3a**). The IP address of this packet is the network broadcast address 255.255.255.255. This address is received by everything on the same network segment. The DHCP server examines the broadcast packet and notes the client's Ethernet address. If the Solaris 9 server is setup as a DHCP relay agent, it will send the contents of the broadcast packet to the DHCP server that is responsible for that network. Some advanced routers can also interpret the packet as a **BOOTP** packet and route the **BOOTP** packet to a DHCP server on another network segment.
2. If a DHCP server only has one Ethernet card installed on one network segment, it uses the appropriate network table to find an IP address for the DHCP client. If a Solaris 9 server has multiple Ethernet cards, it check to see which card the broadcast packet arrived through. This tells the DHCP server which network the packet came from. If the DHCP request came from a DHCP relay agent, the DHCP server looks at the DHCP relay agent's IP address to determine which network the DHCP client resides on.

After determining which network the broadcast packet arrived from, the DHCP server checks its list of IP addresses from a network table. This table has a network number, a list of IP addresses that are being used, and a list of IP addresses that are not being used. The network table and lists of IP address must be set up by the system administrator during the installation of the DHCP server.

Once the server finds an IP address that seems to be unused, it uses the **ping** command to test the network to verify that the IP address is truly not being used.

3. If the candidate IP address is free, the DHCP sends back the DHCP information to the same router (if it came from a DHCP relay agent) or network segment from which the DHCP client made the request. The client looks for its Ethernet address in any broadcast message it sees. Understand that at this point in time, the DHCP client does not have any network configuration information. The DHCP client and the DHCP server must communicate through broadcast messages until the client can configure its network segments.

If multiple DHCP servers have been installed, there is a possibility that the DHCP client can contact several DHCP servers through its broadcast message. This is rare, because most network engineers are very careful to have only one DHCP server assigned to a network segment. On the rare occasions when a client receives multiple DHCP packets from different servers, it will select the first DHCP packet that it receives.

4. While the client is reconfiguring itself with the new network information, the DHCP server places a temporary checkmark next to the assigned IP address. When the client receives its network configuration information, it sends back an acknowledgement packet (**ACK**) back to the DHCP server. If the client never responds back, the DHCP server will eventually make that IP address available for re-use.

If there are two or more DHCP servers on a network segment (rare), one server will receive an acknowledgement and will mark off the IP address from its list of available IP addresses. The other DHCP servers will eventually discard the temporary IP assignment and return the IP address back into its lists of available IP addresses.

5. One of the pieces of information that comes with the DHCP packet is the *lease time* of the IP address. This specifies the amount of time that the network information is valid. At the end of the lease time, the client sends a request for a renewal of the lease to the DHCP server. If the original leasing DHCP server can not be contacted, the DHCP client sends out broadcast messages, looking for another DHCP server to give it an IP address. If a new DHCP lease can not be obtained, the DHCP client disables the Ethernet card that was configured with DHCP information.
6. If the client shuts down, it sends a message to the DHCP server, letting the DHCP server know that it does not need the IP address anymore. This is necessary so that valid IP address are not being held for a computer that might not ever return.

## DHCP Key Points

- A DHCP server can act as a BOOTP relay agent or as a DHCP server, but not both. Either a DHCP server handles DHCP requests, or it passes DHCP requests to other servers.
- A DHCP server can handle a limited number of clients, based upon the database that the DHCP server uses:

|               |                 |
|---------------|-----------------|
| Binary Files  | 100,000 clients |
| NIS+ Database | 40,000 clients  |
| Text Files    | 10,000 clients  |
- Always monitor the network traffic around DHCP servers. DHCP servers and DHCP clients use the 255.255.255.255 network broadcast packet. This packet is broadcast to all network components on a network segment. Try to configure the DHCP server so it is in close proximity to the DHCP clients.
- Never set up any type of server as a DHCP client. DHCP is great for workstations and laptops, but a server needs to have a static IP address. If something happens to the DHCP server, it might send incorrect information that could bring down the entire server structure in a company. Also, there are plenty of software programs, routers and gateways, and that reference a server's IP address. Servers should never be DHCP clients for any reason.

- A DHCP server can not use DHCP itself. A DHCP server must always have a static IP address to ensure the packets can be sent by DHCP relay agents to a DHCP server.
- Be careful when setting up the pool of available IP addresses in a DHCP server. Make sure that the list of available IP addresses does not include the IP address of a server or router or other critical network device!

Imagine what could happen if the static IP address of a server was included in the list of available DHCP addresses. Let's say that a server is taken down for maintenance. Now let's say that during the maintenance cycle, a DHCP server instructed a DHCP client to use that same IP address. When the server or network device came back to life, it would not be able to communicate on the network because its IP address would be in use by something else. To avoid this problem, either disable the IP address in the DHCP server, make the IP address a static address assigned to that device (without a lease expiration), or do not include the IP address in the server's list of available IP addresses.

## WARNING

- The DHCP server in Solaris 9 has a feature that lets this server create client hostnames. This is a very bad feature, which should not be used. Imagine trying to track down hostnames and IP addresses that *both* change. This also causes problems when it comes to network sharing and other specific network settings.
- Do not use DHCP with a server. If something happens to the DHCP server, the client's Ethernet card will be disabled if it can not contact a DHCP server for an IP lease renewal. Why have another server go down with the DHCP server?
- If the number of clients exceeds the number of IP addresses, monitor the DHCP pool. If more than 90% of the IP addresses are leased out, consider increasing the IP address pool.

## DHCP Files

Some critical DHCP run control files are used with DHCP. Only senior level administrators should work with these files. These files are presented here for reference only. If there is a problem with DHCP starting, check to make sure these files exist. Also, understand that if the option Services, Disabled is chosen in the DHCP manager, the DHCP server will not automatically start when Solaris 9 is rebooted.

The run control scripts that relate to DHCP include the following:

- |                                    |                                                                                                                                                                                                                                                                                       |
|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>/etc/rc0.d/K21dhcp</code>    | The shutdown script in the <code>/etc/rc0.d</code> directory that turns off the DHCP server gracefully when the operating system shuts down. The <code>K21dhcp</code> script is also in the <code>/etc/rc1.d</code> directory and the <code>/etc/rc2.d</code> directory.              |
| <code>/etc/rc0.d/K43inetsvc</code> | The shutdown script in the <code>/etc/rc0.d</code> directory that is used to shut down TCP/IP services. This helps to turns off the DHCP server gracefully when the operating system shuts down. The <code>K43inetsvc</code> script is also in the <code>/etc/rc1.d</code> directory. |

- /etc/rc0.d/K90dhcpageant** The shutdown script in the **/etc/rc0.d** directory that turns off the DHCP agent. The DHCP agent is the client side of DHCP that requests information from a DHCP server. This script is used on DHCP clients to gracefully exit when the operating system shuts down. The IP address is released from the client when the server shuts down (provided that the IP address is not reserved).
- /etc/rc1.d/K42inetsvc** A script that works with NIS and NIS+ maps. This script only affects the DHCP server if the DHCP server is working with a NIS or NIS+ database. Among other things, this script releases an Ethernet card from DHCP control.
- /etc/rc2.d/S72inetsvc** A script that works with NIS and NIS+ maps. This script only affects the DHCP server if the DHCP server is working with a NIS or NIS+ database. Among other things, this script sets the variable **dnsservers** to DHCP if DHCP is enabled on a system. This ensures that the workstation gets its network settings from DHCP, and not from a NIS or NIS+ server.
- /etc/rc2.d/S69inetsvc** A script that configures the **sysidnet** used for network parameters.
- /etc/inet/dhcpsvc.conf** A text file in the **/etc/inet** directory that shows the basic configuration of the DHCP server.

Some common variables are:

- DAEMON\_ENABLED=TRUE/FALSE** DHCP server enabled or disabled.
- RUN\_MODE=server/relay** DHCP server set up as a DHCP server or a DHCP relay agent.
- RESOURCE=SUNWfiles** DHCP information is in a text file.
- SUNWbinfiles** DHCP information is in a binary file.
- SUNWnisplus** DHCP information is in a NIS+ map.
- PATH=/path/to/data/source** Location of the DHCP data file.
- CONVER=1** Version of the public module. This variable should not be changed.

## The DHCP Manager

The DHCP server can be configured with command line options, or by using a Java-based GUI known as the *DHCP Manager*. It is important to know how to use command line tools as well as the GUI tool, in case the video display does not work for some reason.

If at all possible, use the DHCP Manager to work with the DHCP server. The DHCP command line commands are extremely picky about syntax and only give vague error messages if something goes wrong.

The DHCP Manager has several functions. The first is to configure DHCP network tables, macros, and options. Other functions including starting and stopping the DHCP server, and enabling or disabling the DHCP server. The configuration of the DHCP server can be exported to a file. This file can then be imported into another DHCP server to duplicate the first server's configuration. The DHCP Manager can also be used to set up the DHCP server as a DHCP relay agent.

One of the benefits of using the DHCP Manager is that the documentation is excellent and there are several installation wizards that guide the administrator through the configuration of DHCP. The installation wizards also check the syntax of the data entered.

A DHCP relay agent is a server that sits on the same network segment as some DHCP clients. When a DHCP client becomes awake, it sends out a broadcast message trying to find a DHCP server. The DHCP relay agent receives the broadcast packets and packages them into IP packets. The IP packet is then sent to a DHCP server over the network. A DHCP relay agent is necessary because the DHCP client can not talk to a DHCP server over the network (because the client does not have an IP address, subnet mask or default router). The DHCP server then sends back the client's configuration information to the DHCP relay agent. The DHCP relay agent sends the information back to the DHCP client. Once the DHCP client has its network configuration set up, the DHCP relay agent is not needed anymore.

One benefit of a DHCP relay agent is that the DHCP server can be centrally located. This makes it easier to manage the DHCP servers in a company. Another benefit is that routers do not have to be set up to pass **BOOTP** packets. If the network topology changes, it's easier to reconfigure a DHCP relay agent than it is to reconfigure routers to handle the network traffic to the DHCP servers. Figure 31.1 shows the DHCP Manager, in which the system administrator can manage the DHCP server with a GUI interface.

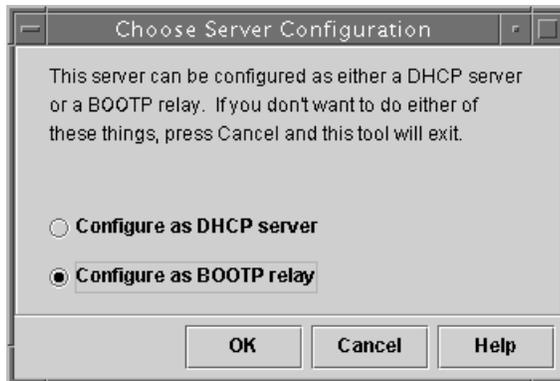


**Figure 31.1 The DHCP Manager**

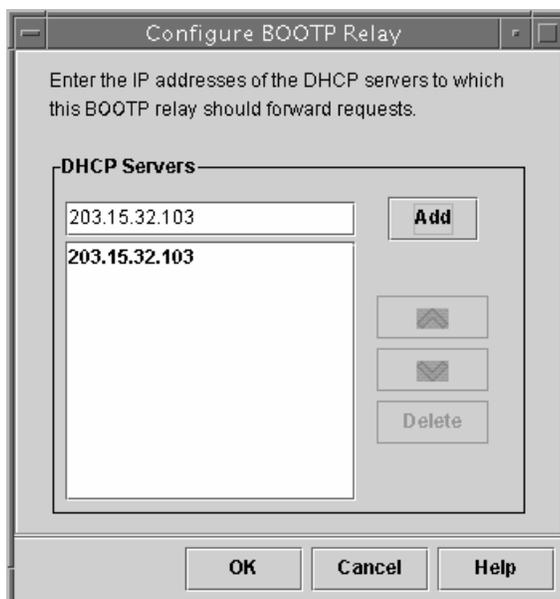
## Lesson 31.1 Set up a Server as a DHCP Relay Agent

This lesson shows how to set up a DHCP relay agent. A relay agent is used to forward DHCP broadcast packets to a specific DHCP server. Understand that with Solaris 9, a DHCP server can only function as a DHCP server or a DHCP relay agent; it can not be both. In this lesson, the DHCP relay agent sends DHCP packets to a fictitious DHCP server. This will cause no harm, as long as the DHCP server is not connected to the Internet (in case there is a device on the IP address used in the lesson).

1. Log in as the root user.
2. Open a Terminal window.  
*To open a terminal window, right click anywhere in unoccupied desktop space. The Workspace menu will appear. Left click on the Tools menu item, then left click on the Terminal window icon.*
3. Type the command `/usr/sbin/dhcpconfig -U -h -x`  
*This command is used to un-configure a DHCP server. If an error message appears that says **dhcpconfig: Error - reading DHCP configuration file. No such file or directory** that only indicates that a DHCP server does not exist.*
4. Type the command `/usr/sadm/admin/bin/dhcpmgr &`  
*The **dhcpmgr** command starts the DHCP manager, the GUI tool used to manage a DHCP server in Solaris 9. The following window should now appear:*
5. Type the command `/usr/sadm/admin/bin/dhcpmgr &`  
*The **dhcpmgr** command starts the DHCP manager, the GUI tool used to manage a DHCP server in Solaris 9. The following window should now appear:*



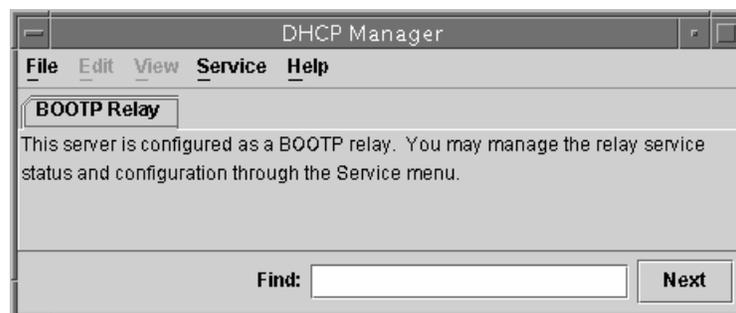
6. Select  "Configure as BOOTP relay" and click OK.  
*The following window should now appear:*



7. Type in a fictitious IP address for a DHCP server, as shown above. Then left click the OK button.

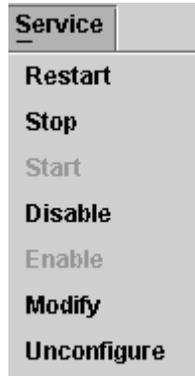
*In the above figure, several menu items are grayed out, because they can not be used on a BOOTP relay agent. This is a rather chopped down version of the DHCP manager. This DHCP manager only has configuration information for the DHCP relay agent.*

*The following window will now appear:*



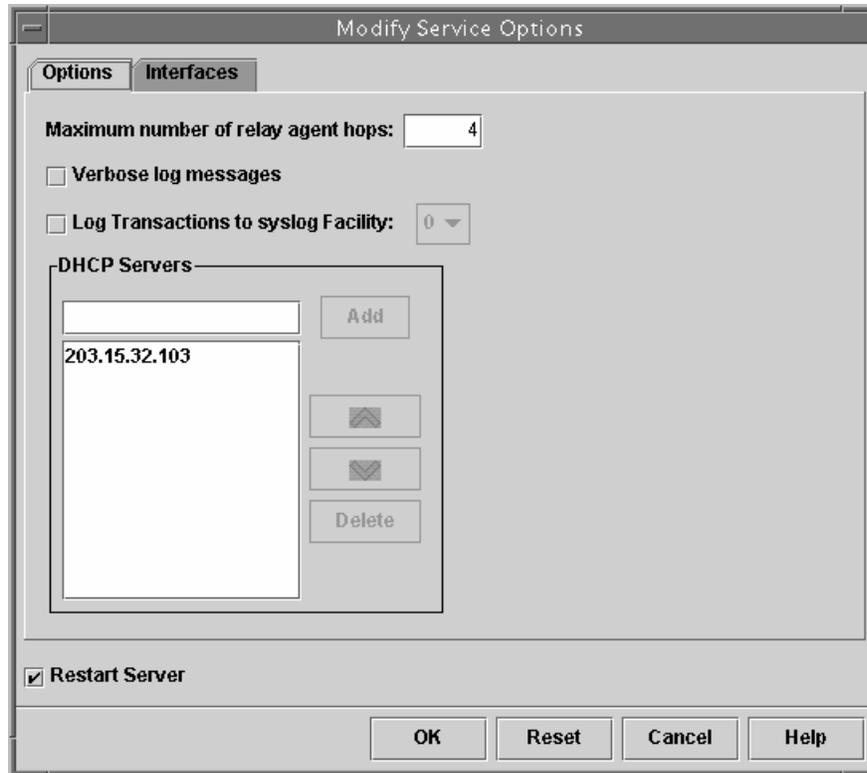
8. Left click on Service in the Menu bar.  
*The Find text box at the bottom of the screen is used to find a particular IP address. This is a useful troubleshooting tool. Experiment with this feature if you want to; it can not harm your system or network.*

*The Service menu has several options that should be understood:*



- Restart - Force the **in.dhcpd** daemon to re-read all of its configuration information*
- Stop - Stop the DHCP **in.dhcpd** daemon*
- Start - Start the DHCP **in.dhcpd** daemon*
- Disable - Don't start the DHCP server when Solaris 9 starts or when Solaris 9 is rebooted*
- Enable - Start the DHCP server when Solaris 9 starts or when Solaris 9 is rebooted*
- Modify - Modify the DHCP relay agent*

9. Left click on Modify.  
*The Modify Service Options window will appear.*
10. Left click on the Options tab.  
*The window shown in Figure 31.2 will appear.*



**Figure 31.2 Options Tab of the Modify Service Options Window**

The first selection item is "Maximum number of relay agent hops:" *This indicates the maximum number of times the DHCP relay agent will try to contact a DHCP server before it quits.*

- Leave the default value of 4 alone.
- Do not select the  "Verbose log messages" or  "Log Transactions to syslog Facility" choices.

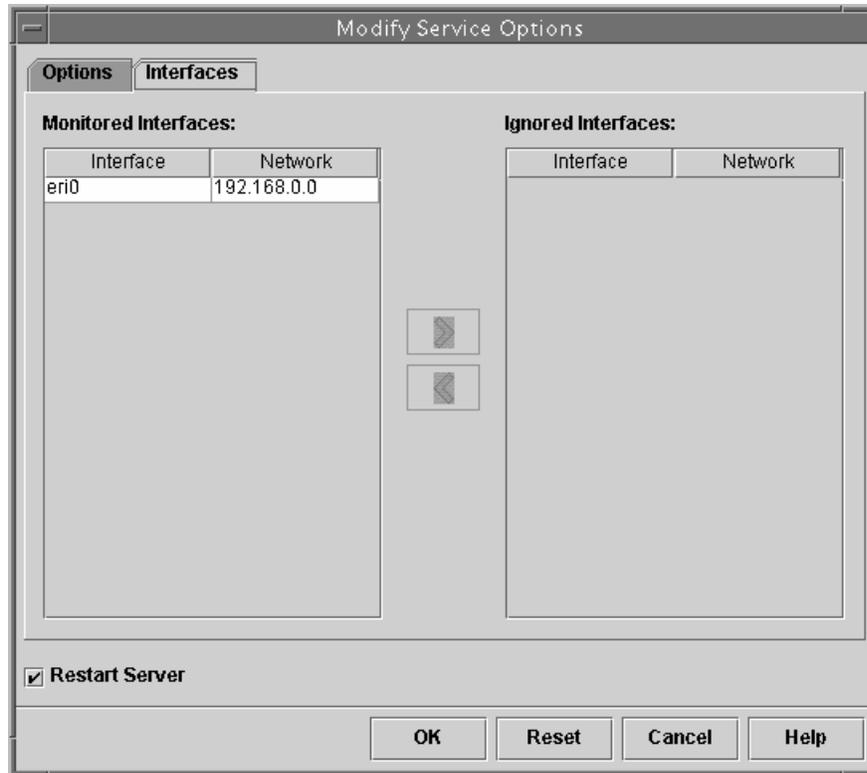
11. Make sure that the checkbox  "Restart Server" is checked (it is at the lower left of the window).

*Then left click on OK.*

*Anytime a change is made to the DHCP server, restart the server. This only restarts the DHCP server, not the operating system. Because DHCP clients try several times to contact a DHCP server, there should not be any significant DHCP problems during a restart of the DHCP from the DHCP manager.*

12. Left click on the Interfaces tab.

*The window shown in Figure 31.3 will appear. This window shows what Ethernet cards and network segments the DHCP relay agent should monitor (or ignore) for DHCP client broadcast messages. For servers with only one Ethernet card, this option really does not apply. For servers with multiple Ethernet cards, this is a convenient way to monitor different networks.*



**Figure 31.3 Interfaces Tab of the Modify Service Options Window**

13. Left click on the OK button.  
*Feel free to explore the DHCP relay agent, but do not make any permanent changes to the system. When you are done, click on Cancel so that any changes you have made are not permanent.*
14. When you are finished, left double click on the Windows Menu button on the top right corner of the window. It looks like a dash inside a small square.

## Lesson 31.2 Creating a DHCP Server with the DHCP Manager

### **WARNING**

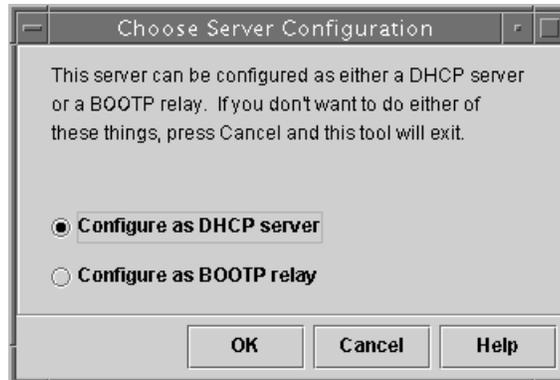
If your test system is on a live network, it could take DHCP requests from company clients. Make sure that the network segment is isolated from the rest of the network. If this system can not be disabled, make sure there are no other DHCP clients on the network segment, and that the router (which separates different network segments) has BOOTP disabled for this network segment. If DHCP is configured for this network segment, it will entertain DHCP requests.

This lesson focuses on the DHCP Manager. The DHCP manager is a Java-based GUI tool for managing the DHCP server. In this lesson the DHCP relay agent created in the previous lesson will be destroyed. The DHCP manager will then be used to set up this system as a DHCP server.

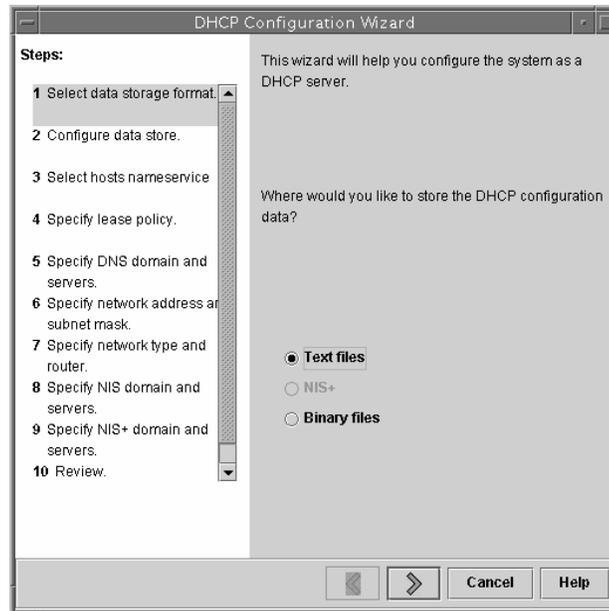
1. Log in as the root user.
2. Open a Terminal window.
3. Type the command `dhcpcconfig -U -h -x`

This command is used to un-configure a DHCP server. Don't worry if an error message appears that says **dhcpconfig: Error - reading DHCP configuration file. No such file or directory**; this only indicates that a DHCP server does not exist.

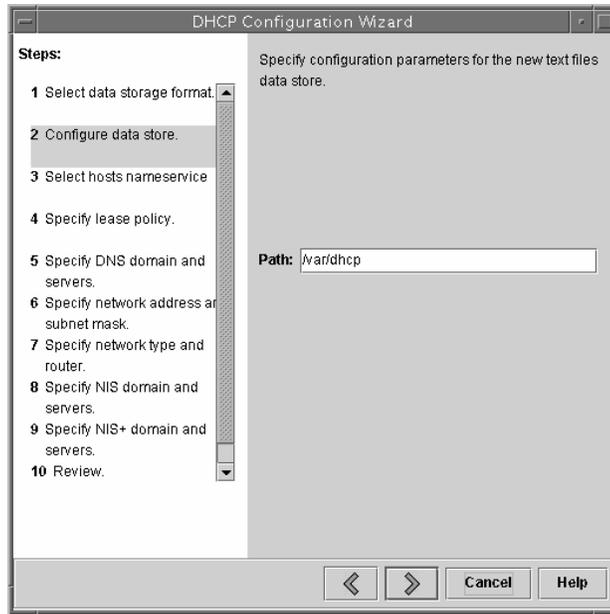
4. Type the command `/usr/sadm/admin/bin/dhcpmgr &`  
When the DHCP server starts, if and DHCP has never been set up, the following window pops up:



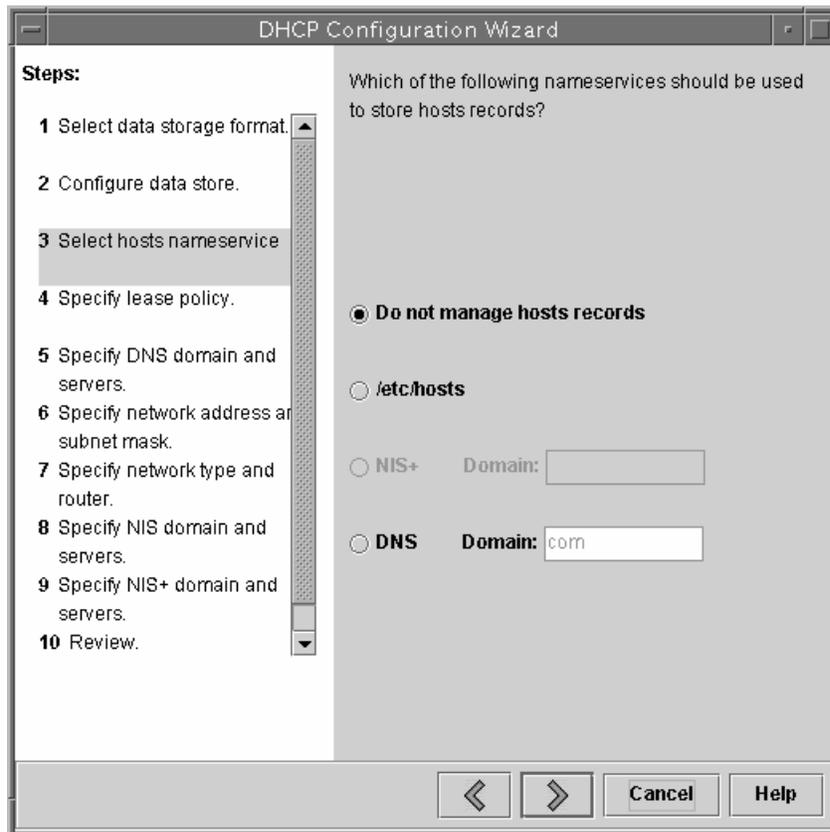
5. Select  "Configure as DHCP server" and left click on OK.  
The DHCP Configuration Wizard starts. The first screen asks for the Data storage format. If NIS+ is not enabled, the NIS+ option is grayed out



6. Select  "Text files" and left click on **>**  
The next screen asks for the path location for the DHCP files. This is only asking for the directory structure.



7. Just click on **>**. Don't add any file names to this path: **/var/dhcp** is the default option. The third screen asks which name service should be used to store host records. A name service matches IP addresses to a hostname.

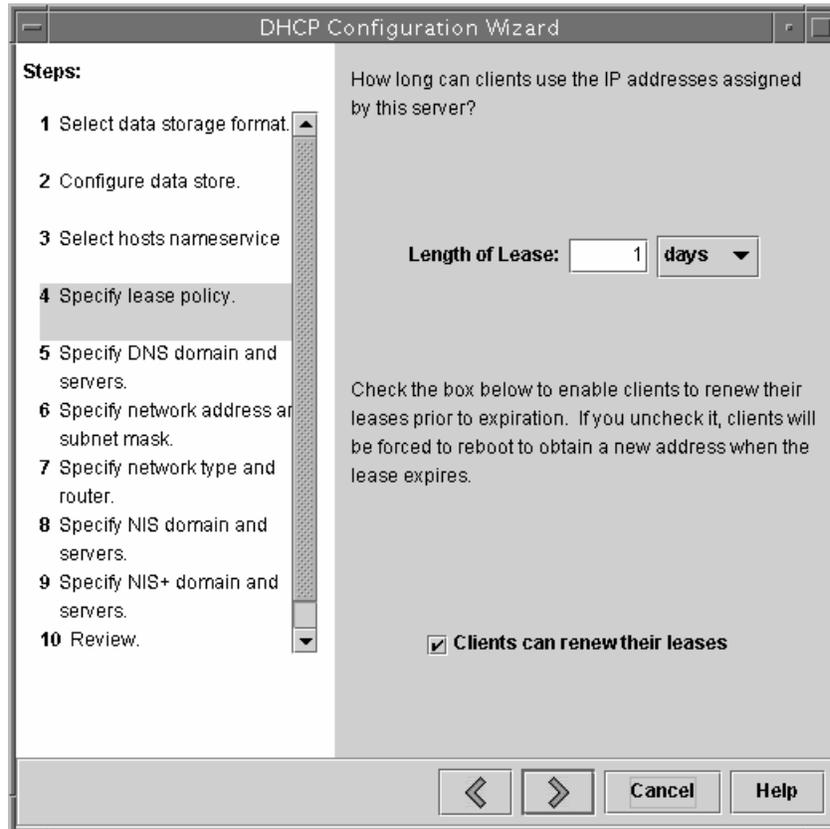


8. Select **Do not manage hosts records** and click on **>**.  
Chapter 31 DHCP

**Notes:** (1) If you have a very small network or are just working with a test workstation, you could select  /etc/hosts. When this option is selected, all configuration changes are saved in the /etc/hosts file.

(2) If you have a large network, you could select  DNS (provided that the DNS server can work with DHCP).

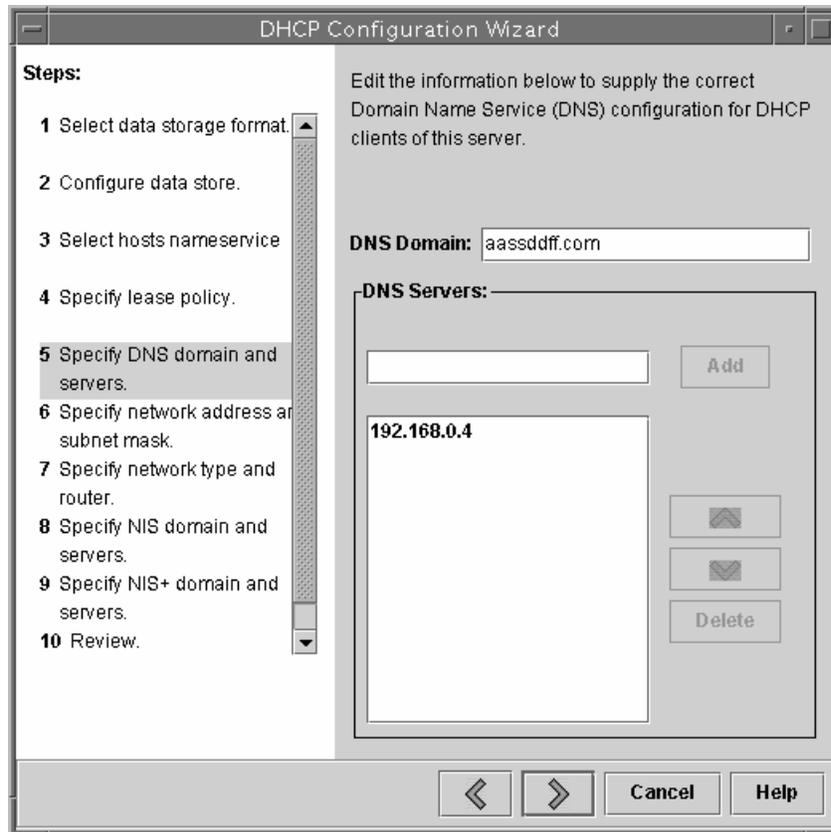
The next box sets the amount of time a client can have an IP address before the lease has to be renewed.



9. Fill in this window as follows:
  - a. In the "Length of Lease" text box, enter **1**.
  - b. Make sure that the checkbox  "Clients can renew their leases" is checked.
  - c. Click on .

*It is not a good idea to require clients to reboot to obtain a new address. ISPs (Internet Service Providers) usually require this option to prevent customers from using DHCP to set up an IP address and creating a server from that IP address.*

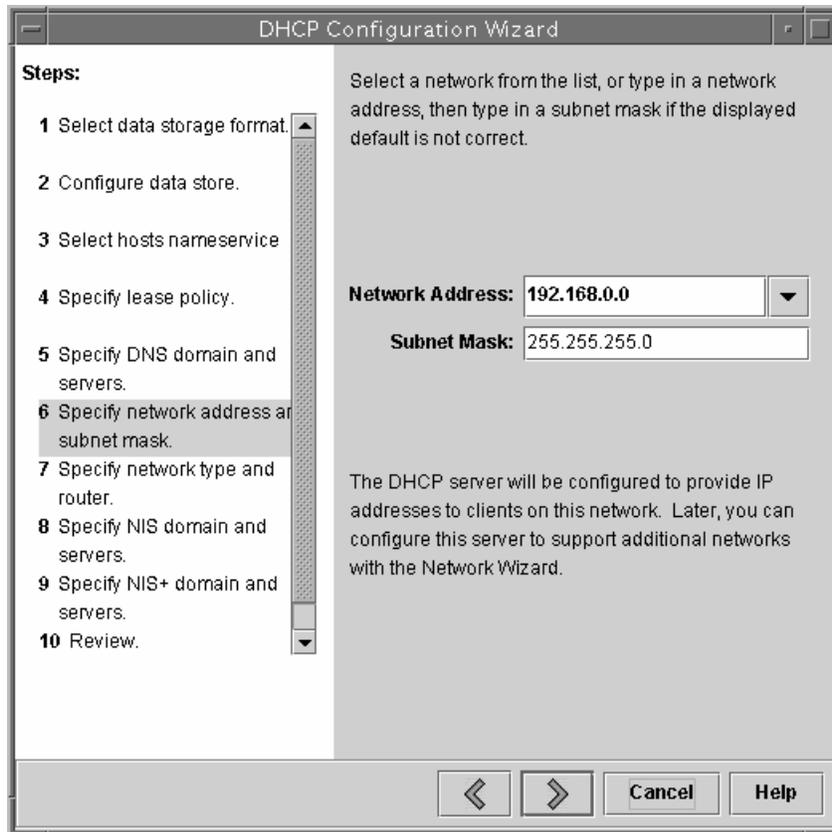
*The fifth screen asks for the name of DNS servers, among other information:*



10. Fill in this window as follows:

- a. In the DNS Domain text box, type in your domain name (if you have one). If you do not have a domain name, leave this text box blank.
- b. For the DNS Servers area, proceed as follows:
  - If your network can connect to a DNS server, in the DNS Servers area, type the IP address of any DNS servers that are used to resolve hostnames to IP addresses for this workstation. Then left click on the Add button so that the IP addresses are recorded.  
*This DNS information will be passed on to DHCP client workstations.*
  - If your network cannot connect to a DNS server, leave this area blank.
- c. Click the >button.

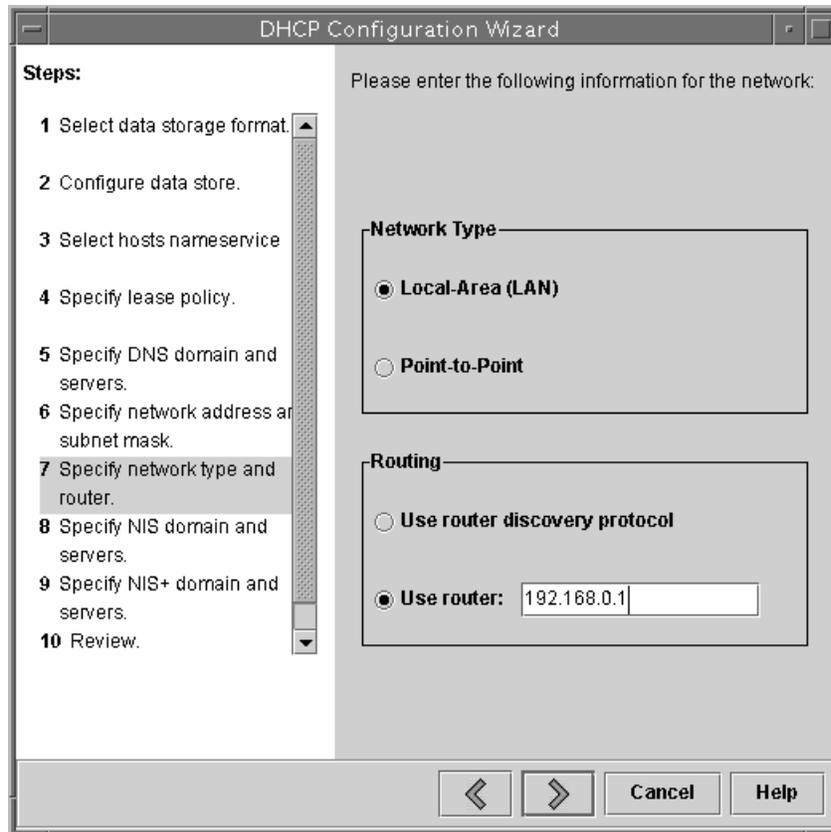
*The sixth screen asks the user for the Network Address and Subnet Mask that the DHCP server will work with.*



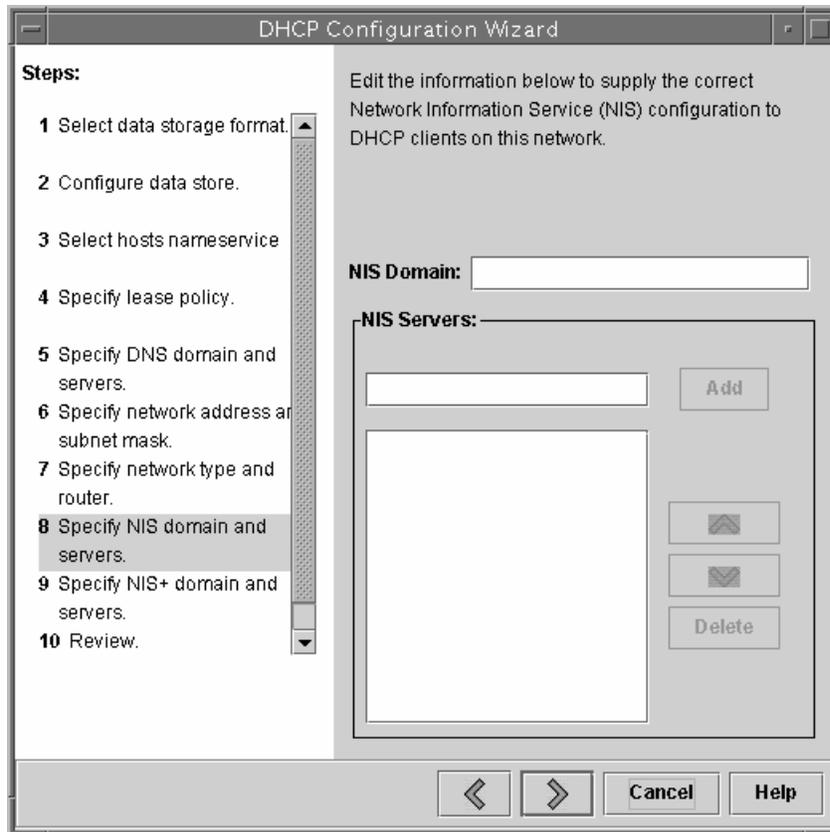
11. This window will try to display the correct Network Address and Subnet Mask, based upon file that it finds in the */etc* directory.

- If these values are correct, click the ➤ button and continue with the next numbered step.
- If these values are not correct:
  - a. In the Network Address text box, select a network list, or type in a Network Address. If you have followed the previous lessons in this book type **192.168.0.0** here.
  - b. In the Subnet Mask text box, type in the appropriate Subnet Mask, as follows:
    - Class A Subnet: 255.0.0.0
    - Class B Subnet: 255.255.0.0
    - Class C Subnet: 255.255.255.0
    - Variable length or non-standard subnet mask (a mask that uses any number other than 0 and 255), consult network subnet tables to determine your network address.
  - c. Click the ➤ button.

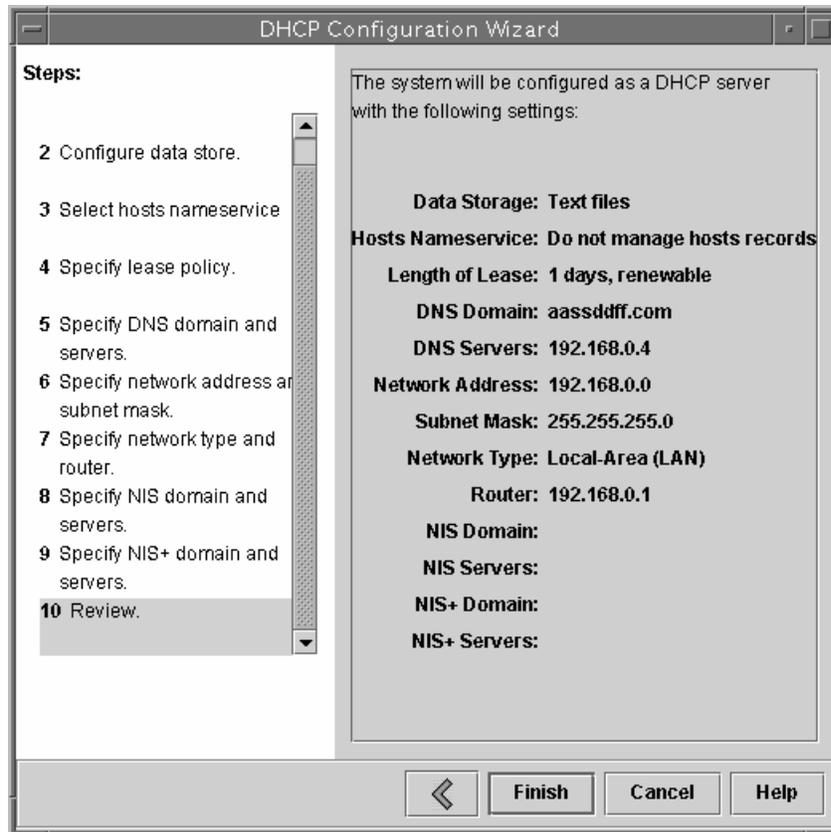
*The seventh screen asks for the Network Type. This can be a Local-Area Network (LAN) or Point-to-Point. If the computer is attached to the network, it is on the LAN. If the computer connects through a modem or other terminal line service, it connects via Point-to-Point. It is also possible to use the router discovery protocol or to specify a router's IP address.*



- In this example, specify a router.  
*The eighth and ninth screens ask for the location of NIS and NIS+ domain servers. These are NIS and NIS+ servers for the DHCP server only.*



13. Leave these fields blank in both screens. Just left click on the ➤ button.  
*The tenth screen lets you review the information that you have entered.*



14. Carefully read the information in the right panel.

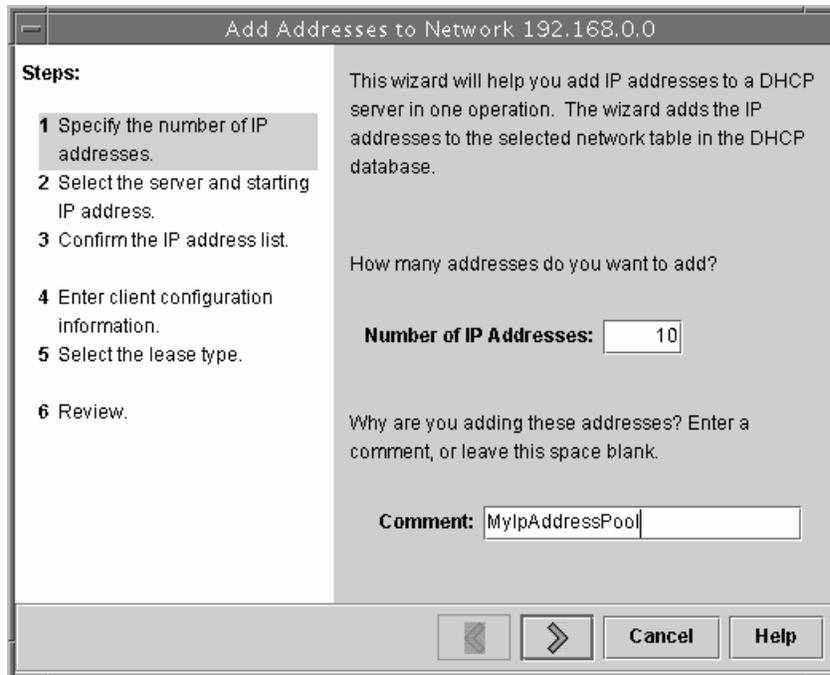
- If any information is incorrect, left click on the < button as needed to reach the screen for entering that information. After correcting the information, use the OK or > button on each screen to navigate back to the screen shown above.
- If all the information is correct, left click on the Finish button.  
*After the DHCP server has its initial settings, it still needs to have the pool of IP addresses specified. The previous information only told the DHCP server what type of network it was on.  
 When the DHCP Manager starts again, a popup window appears as shown below:*



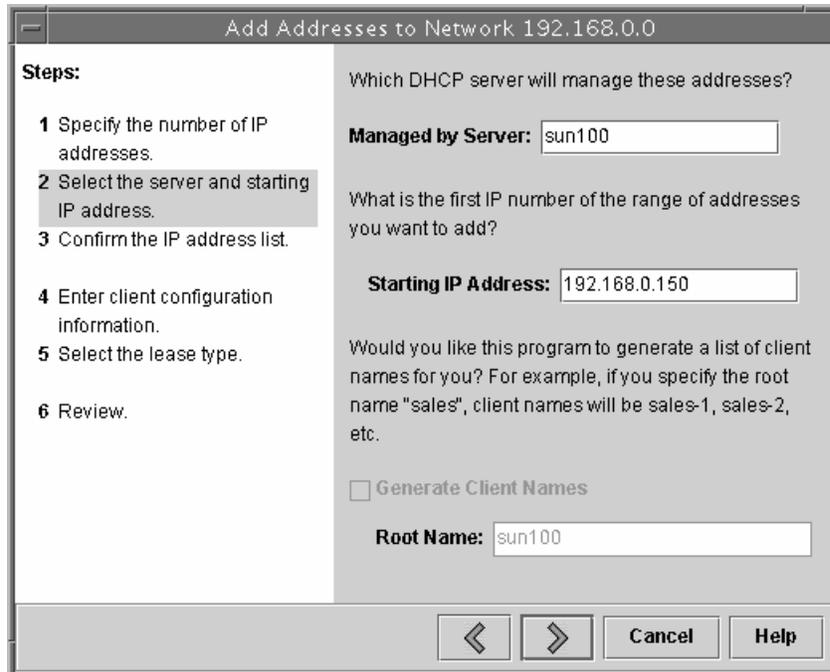
15. Press the Yes button.

*This wizard has six steps*

*The first window asks for the number of IP addresses and for a comment for this IP address pool.*



16. In the Number of IP Addresses box, type **10**.  
 In the Comment box, type **myIPAddressPool**  
*The second screen asks which DHCP server will manage these addresses.*



17. Fill out this screen as follows:
- a. In the Managed by Server text box, enter the hostname or IP address of this system. This is the DHCP server.

- b. In the Starting IP Address text box, enter the Starting IP address.  
*It would most likely be safe to use **192.168.0.150**, for the reasons noted below.*

**Note:** *Only the starting IP address is requested. This is used as the starting point for generating the number of IP addresses requested in the previous step. For example, if you specified 10 IP addresses there and specified a starting address of 192.168.0.150 here, you would be allocated the addresses from 192.168.0.150 through 192.168.0.159.*

*The starting IP address depends on your network topology. Most VPN networks that use the 192.168.x.x number scheme don't use IP addresses above **192.168.0.100**. Most routers and ISPs reserve the use of numbers less than 100 for internal use and dedicated functions. The address 192.168.0.150 should be a safe one to use.*

*To see if an IP address is already in use, open a Terminal session and type*  
**ping <IP\_address\_to\_check>**

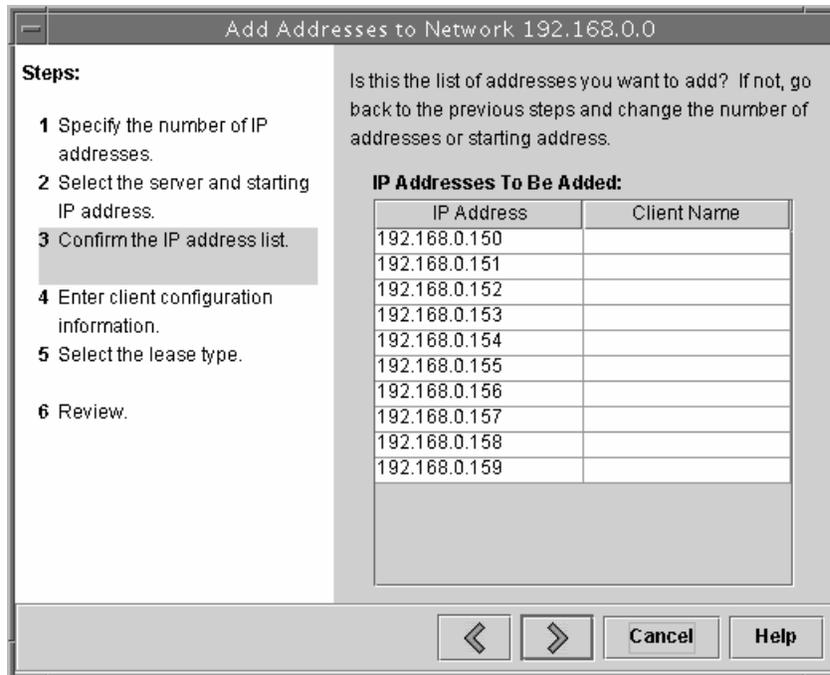
*Some networks disable ping responses to prevent ping denial of service attacks, so this might not be an accurate way of detecting IP addresses. You can also try the following commands:*

**telnet <IP\_address\_to\_check>**  
**ftp <IP\_address\_to\_check>**  
**spray <IP\_address\_to\_check>**

*Most likely the device will respond to one of these commands.*

- c. Make sure that the  Generate Client Names box remains unchecked.  
*Do not let the DHCP server generate client names! Allowing it to do so can create a real problem when it comes to troubleshooting! It's hard enough to track down a hostname with DHCP, but trying to track down both a hostname and an IP address would be extremely difficult.*
- d. Click on ➤

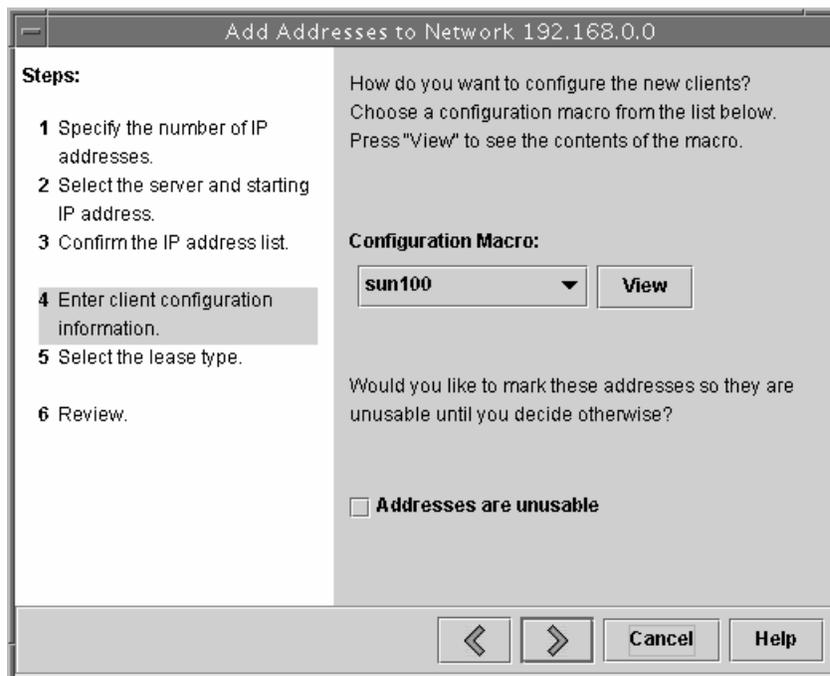
*The third screen shows the DHCP addresses that were created.*



18. If the list of IP addresses is correct, click > and go to the next step. If the list is incorrect, click < and repeat the previous step.

*The fourth screen shows the Configuration Macro that will be associated with the pool of IP addresses. A macro has all the network settings related to DHCP. These settings can be changed at a later time.*

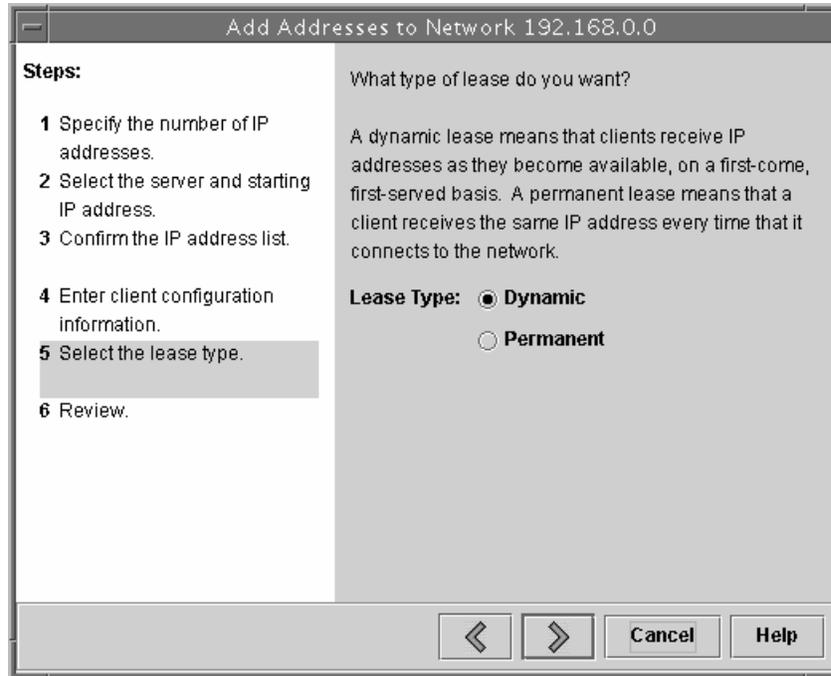
*Notice the checkbox that says “Addresses are unusable.” This is used when network servers and routers require static IP addresses. This pool of IP address is in DHCP, but nothing can use these IP addresses.*



19. Select the configuration macro that matches the system’s hostname and then click the > button.

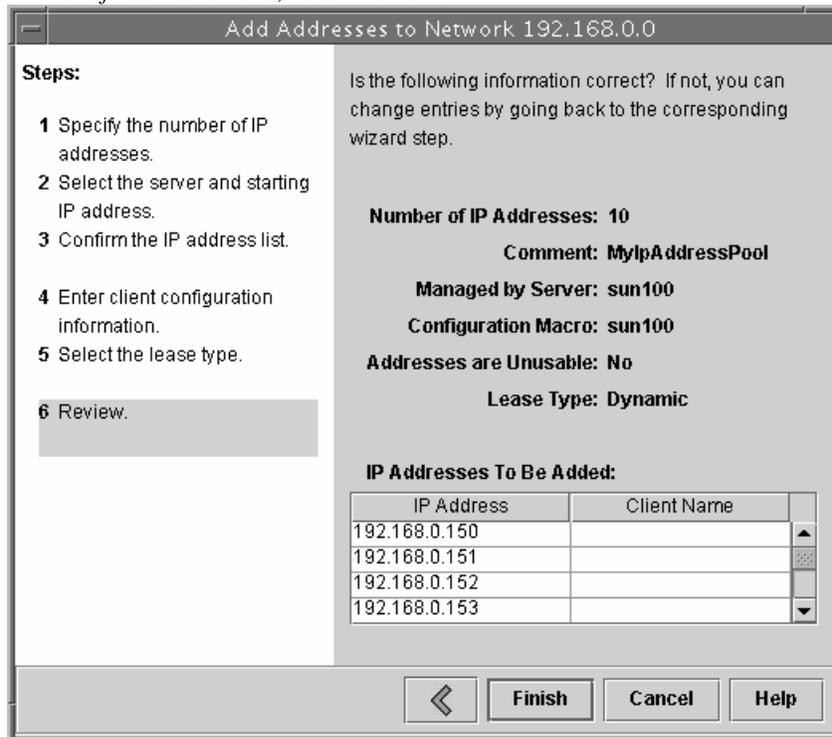
If a macro with the system's hostname is not visible, click on the ▼ button and then choose that macro. Leave the  "Addresses are unusable" box unchecked.

The fifth screen asks what type of DHCP lease should be presented.



20. Click on  Dynamic.

The last screen is a confirmation screen, as shown below.



- If all the information is correct, left click on the Finish button.

21. In this screen:

- If any information is incorrect, left click on the ◀ button as needed to reach the screen for entering that information. After correcting the information, use the OK or ▶ button on each screen to navigate back to the screen shown above.
- If all the information is correct, left click on the Finish button.

*For some reason, the DHCP server is really slow to create the DHCP addresses. If the screen seems to freeze, don't shut down the server or the DHCP manager. Just wait a short period of time. The DHCP manager should appear again.*

## Configuring a DHCP Client

Solaris 9 can be a DHCP client or a DHCP server, but it can not be both at the same time. A DHCP server needs to have a static IP address for itself. To make Solaris 9 system operate as a DHCP client, follow the steps in Lesson 31.3.

### Lesson 31.3 Creating a DHCP Client

**Note:** To follow this lesson, you must have more than one SPARC workstation. This lesson is impossible with only one workstation, because a DHCP server can not also be a DHCP client.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command `ifconfig -a`  
*This will show all the network interfaces on the system. The Ethernet card's network\_interface name is the first word on the left side of the screen. Do not use the `lo0` interface or any interface that starts with `lo`. The `lo` interface is a loopback 127.0.0.1 virtual interface that can not be configured for DHCP.*
4. Type the command `touch /etc/dhcp.<network_interface>`  
*Examples: touch /etc/dhcp.hme0  
touch /etc/dhcp.eri0*
5. Type the command `/usr/sbin ifconfig <network_interface> dhcp start`  
*Examples: /usr/sbin ifconfig hme0 dhcp start  
/usr/sbin ifconfig eri0 dhcp start*
6. Type the command `/usr/sbin/ifconfig -a`  
*The Ethernet card should take its network settings from the DHCP server. If for some reason the workstation does not reconfigure itself from the DHCP server, reboot the workstation. Once the workstation starts again, type the command `/usr/sbin/ifconfig -a` to make sure the new network settings are from the DHCP server.*

## Quick Tip

Microsoft Windows is compatible with the Solaris 9 DHCP server. To use a Microsoft Windows system as a DHCP client:

1. Set up the Windows system to “Obtain an IP address automatically.”
2. Reboot the Windows system.
3. Open an MS-DOS window.
4. Type the command `ipconfig -all`  
This should show the IP address, subnet mask and other DHCP entries that were set up by the server.
5. Type `exit` to return to Windows from the MS-DOS prompt.

**Note:** You can also use the Windows command `winipcfg` to see these settings in a GUI window. Use the Release All and Renew All buttons to see DHCP in action.

### Understanding Macros and Options

Once a DHCP pool of IP address has been created, it needs to have a macro attached to it. The macro contains the information that is assigned to the client. For example, the macro created in the last lesson sends the following variables and their values to a DHCP client

|           |             |
|-----------|-------------|
| Include   | Locale      |
| Timeserv  | 192.168.0.4 |
| LeaseTim  | 86400       |
| LeaseNeg  |             |
| DNSdomain | aassdff.com |
| DNSserv   | 192.168.0.4 |

As you can see, a macro is a collection of settings for **Options** = variables, such as **Timeserv**, **LeaseTim** and **LeaseNeg**.

A macro's name determines which client or clients will receive the information. For example:

| Macro Name   | Clients That Receive the Information      |
|--------------|-------------------------------------------|
| 192.168.0.5  | The client with                           |
| 192.168.0.0  | All clients on the 192.168.0.0 network    |
| SUNW,sun4u   | All clients with the sun4u architecture   |
| 08002032B3A4 | The client with this specific MAC address |

Some of these macros may not exist. For example, a system administrator could only make a Network Address macro and not make any others.

There are four official Macro Classes:

|                       |                                     |
|-----------------------|-------------------------------------|
| Client Class Macro    | - Clients with the same hardware    |
| Network Address Macro | - Clients on a particular network   |
| IP Address Macro      | - Client with a specific IP address |

Ethernet Address Macro - Client with a specific Ethernet MAC address

These macros are processed in following order:

- 1 - Client Class
- 2 - Network Address
- 3 - IP Address
- 4 - Ethernet Address.

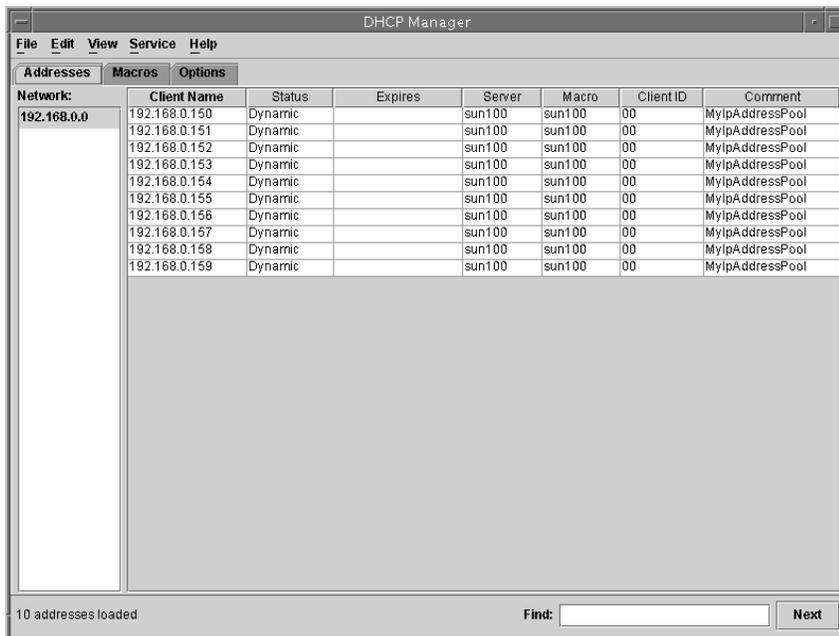
All the macros that apply to a client are processed. If an option is in more than one macro, the last macro that contains the option will set the option's value.

## Lesson 31.4 Configuring Macros And Options

Now that the DHCP server has been created with 10 IP addresses, those DHCP clients are using a rather limited set of macros and options. Remember that a Macro holds all the configuration information for a client, such as DNS domain, the DNS server's IP address, and a Web server's IP address. Options that have been configured, such as Timeserv and LeaseTim, are passed to the DHCP client.

In this lesson the reader will add extra options to a macro. The first option, WWWservs is selected and given an IP address. The reader then creates a fictitious option.

Figure 31.4 shows a configured DHCP server.

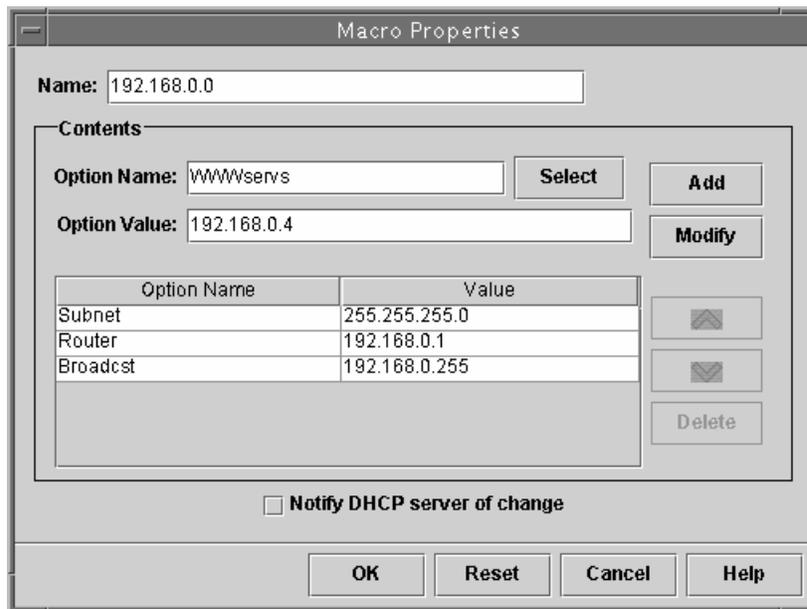


| Network:    | Client Name   | Status  | Expires | Server | Macro  | Client ID | Comment         |
|-------------|---------------|---------|---------|--------|--------|-----------|-----------------|
| 192.168.0.0 | 192.168.0.150 | Dynamic |         | sun100 | sun100 | 00        | MylpAddressPool |
|             | 192.168.0.151 | Dynamic |         | sun100 | sun100 | 00        | MylpAddressPool |
|             | 192.168.0.152 | Dynamic |         | sun100 | sun100 | 00        | MylpAddressPool |
|             | 192.168.0.153 | Dynamic |         | sun100 | sun100 | 00        | MylpAddressPool |
|             | 192.168.0.154 | Dynamic |         | sun100 | sun100 | 00        | MylpAddressPool |
|             | 192.168.0.155 | Dynamic |         | sun100 | sun100 | 00        | MylpAddressPool |
|             | 192.168.0.156 | Dynamic |         | sun100 | sun100 | 00        | MylpAddressPool |
|             | 192.168.0.157 | Dynamic |         | sun100 | sun100 | 00        | MylpAddressPool |
|             | 192.168.0.158 | Dynamic |         | sun100 | sun100 | 00        | MylpAddressPool |
|             | 192.168.0.159 | Dynamic |         | sun100 | sun100 | 00        | MylpAddressPool |

**Figure 31.4 DHCP Manager With 10 Addresses**

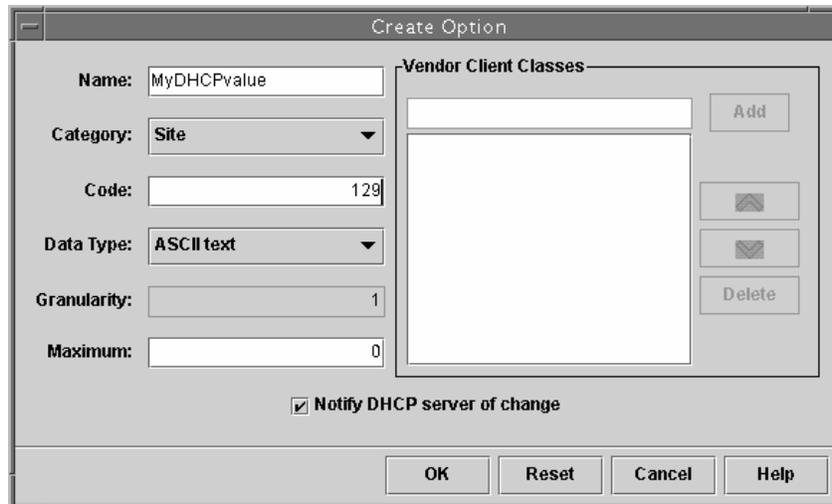
1. Log in as the root user.
2. Start the DHCP manager with the command  
`/usr/sadm/admin/bin/dhcpmgr &`
3. Left click on the Addresses tab.  
*The macro assigned to this IP address is listed in the Macro tab. The macro on the author's computer is sun100. The hostname of the author's computer is sun100.*
4. Left click on the Macros tab.

5. Left click on Edit.
6. Left click on Properties  
*The Macro Properties window should appear, as shown in Figure 31.5.*



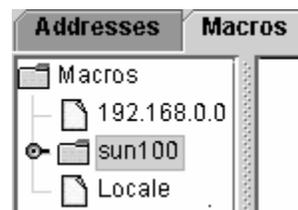
**Figure 31.5 Macro Properties Window**

7. Left click on the Select button next to the Option Name: text box.
8. Scroll down until the option WWWservs is displayed.
9. Left click on OK.
10. In the Option Value text box, type the IP address of your computer.
11. Left click on the Add button.
12. Left click on OK.  
*The final results should look like Figure 31.5.*
13. Left click on the Options tab.
14. Left click on Edit in the Menu bar.
15. Left click on Create.  
*The Create Option window should appear, as shown in Figure 31.6*

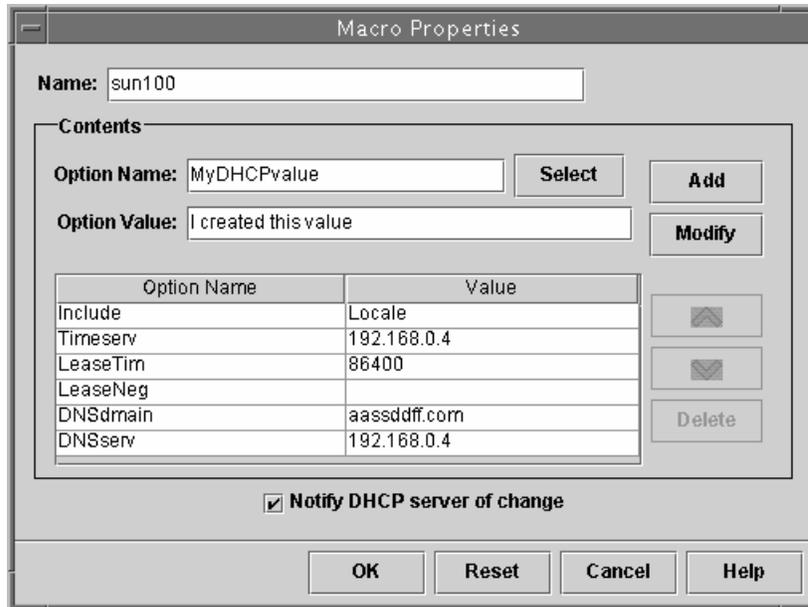


**Figure 31.6 Create Option Window**

16. In the Name: text box, type **MyDHCPvalue**.  
*This step and the next one simply create a dummy option called **MyDHCPvalue** with a dummy value of **129**. The point is simply to enter an option and its value.*
17. In the Code: textbox type **129**
18. In the Data Type: field, select ASCII text.
19. In the Maximum text box, type **0** (zero).
20. Make sure that the checkbox  "Notify DHCP server of change" is checked.
21. Left click on OK.  
*From here to the end of the lesson, we will insert the **MyDHCPvalue** option into the sun100 macro.*
22. Left click on the Macros tab.
23. Highlight the Sun100 macro (or whatever macro has the same name as your system's hostname), as shown below.



24. Left click on Edit
25. Left click on Properties.  
*The Macro Properties window should now appear, as shown in Figure 31.7.*



**Figure 31. 7 Macro Properties Window**

26. Left click on the Select button next to the Option Name: text box.  
*The Select Option popup window should appear.*
27. Left click on Category: at the top of the window.
28. Select Site: as the category.  
*The **MyDHCPvalue** option should appear.*
29. Highlight the **MyDHCPvalue** option.
30. Left click on OK.  
*The Macro Properties window should appear again*
31. In the Option Value: text box, type **I created this value**  
*The Macro Properties window should now look like Figure 31.7.*
32. Left click on Add.
33. Left click on OK to close the window.  
*The macro should now have the custom DHCP Option “MyDHCPvalue”*

### **Importing and Exporting DHCP Configuration Information**

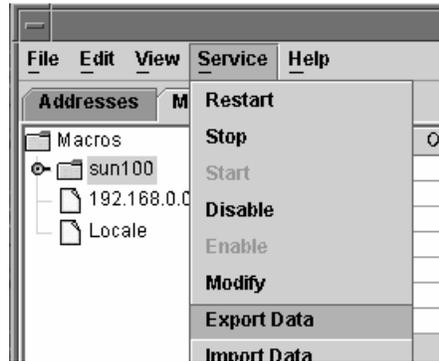
The DHCP manager has a feature that lets you export a DHCP server's settings to a file. That file can then be used to help set up a new DHCP server. This is a nice option if a DHCP server is being replaced by another one (such as a new and more powerful server). The old DHCP server's configuration file can be copied to the new server through a network connection. When the old server is taken off line, the new server can be started with the exact same configuration. This provides for seamless migration from one DHCP server to another one.

### **Lesson 31.5 Exporting DHCP Server Information**

In this lesson, readers will save the configuration of the DHCP server to a file named **myexportfile** in the root ( / ) directory. This file could then be used to set up another DHCP server identically to this one.

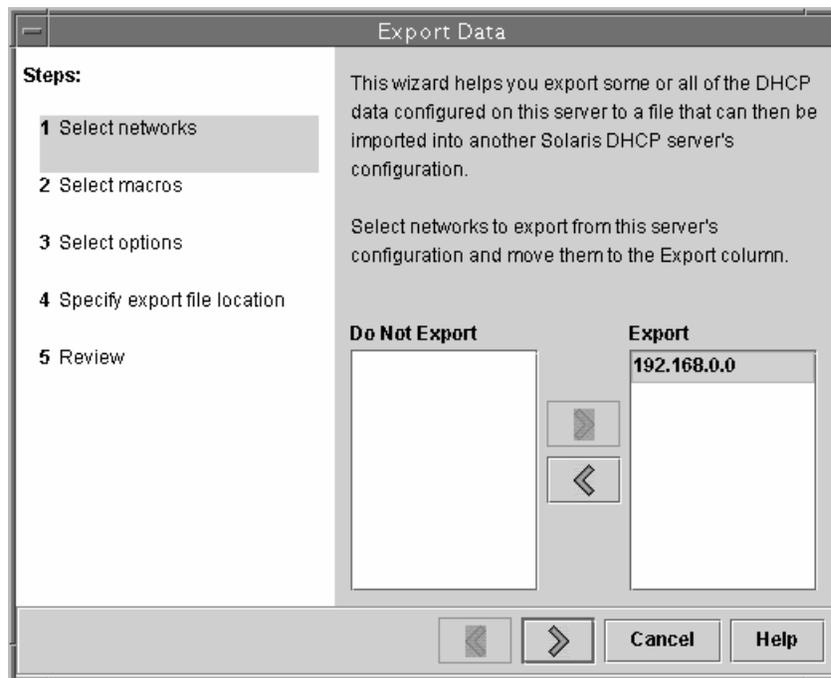
1. Log in as the root user.
2. Open a Terminal window.
3. Start the DHCP manager with the command:  
`/usr/sadm/admin/bin/dhcppmgr &`

- In the first part of this lesson, we will export data from a DHCP server, using the Export Data wizard.*
4. Left click on Service and then left click on Export Data (as shown below).

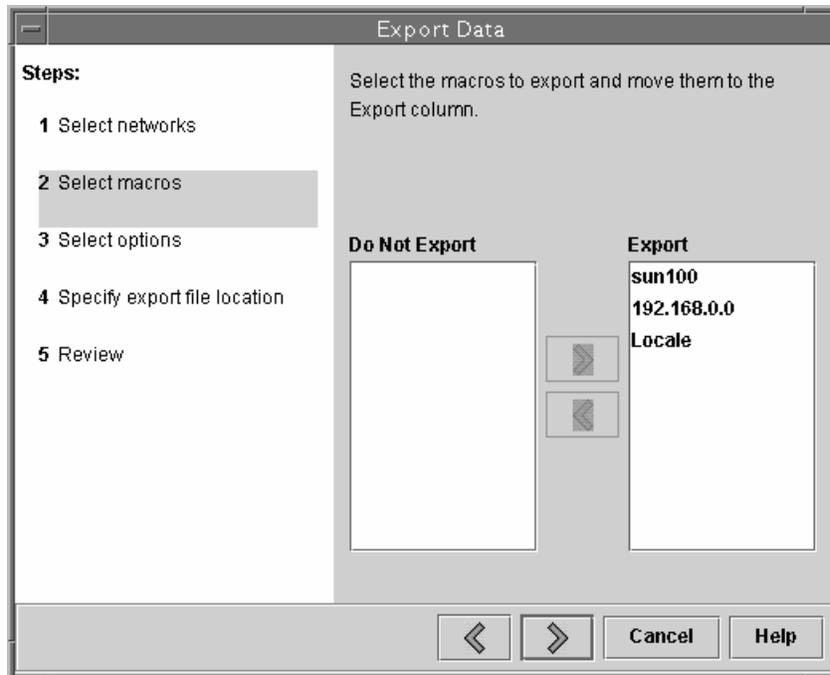


*The first screen asks what network should be exported. If you have followed the lessons in this book, there will be only one choice: 192.168.0.0.*

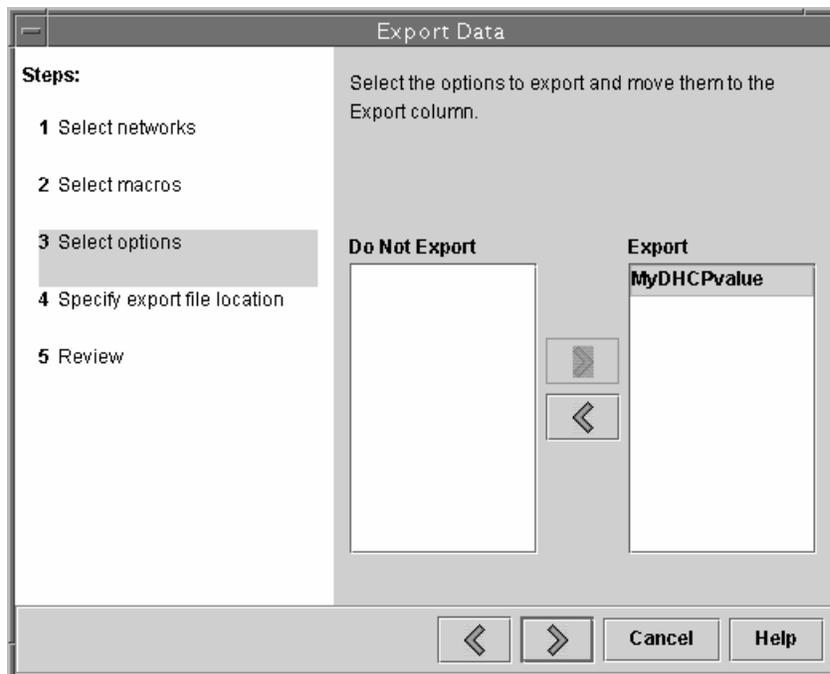
5. In the Do Not Export field, highlight the 192.168.0.0 entry.
6. Left click on the > button to the right of the Do Not Export field, to move that entry to the Export field.
7. Left click on the > button at the bottom of the screen.



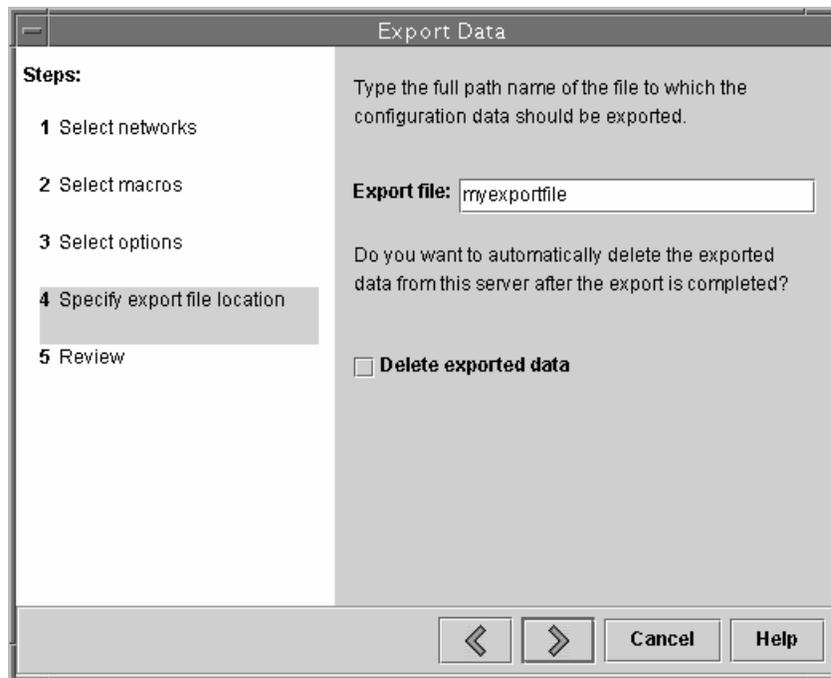
*The next step lets you choose which macros to export. Remember, a macro contains a particular set of configured options.*



8. In the Do Not Export field, highlight all the macros you want to export.
  9. Left click on the ➤ button next to that field, to move those macros to the Export field.
  10. Left click on the ➤ button at the bottom of the screen.
- The next window lets you select which options should be exported.*



11. In the Do Not Export field, select the MYDHCPvalue.
  12. Left click on the ➤ button next to the Do Not Export field to move it into the Export field.
  13. Left click on the ➤ button at the bottom of the screen.
- The fourth window asks for the name of an export file.*



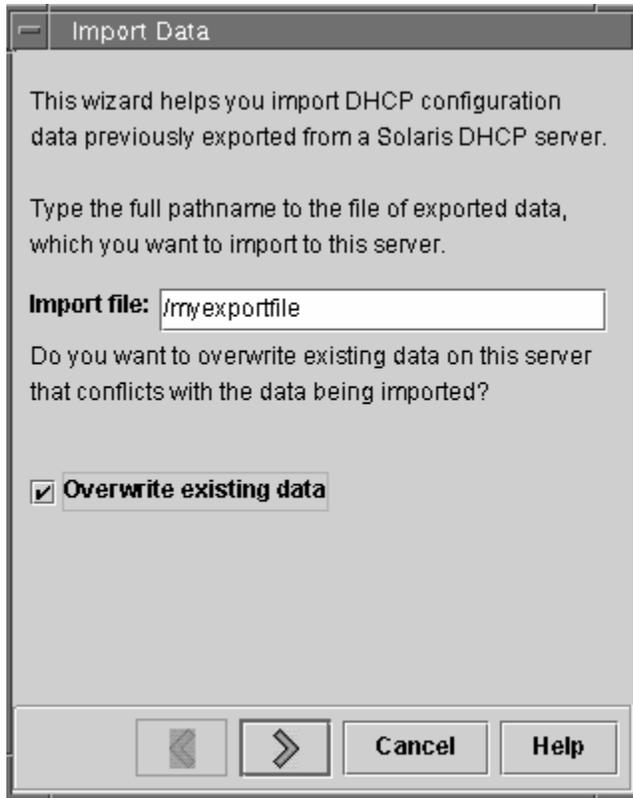
14. In the Export File: text box, type **myexportfile**.
15. Make sure that the  "Delete exported data" checkbox is checked.  
*The final screen is a confirmation screen.*
16. Left click on the Finish button.
17. Left click on **F**ile and then on **E**xit to exit

## Lesson 31.6 Importing DHCP Server Information

As mentioned before Lesson 31.5, after you have exported a DHCP server's settings its settings to a file, The that server's configuration file can be copied to a new server, usually through the a network connection. This provides for seamless migration from one DHCP server to a new DHCP another one.

This lesson shows how to read in (import) the exported configuration information. Because you will be reading it back in to the same server from which you exported it, no changes will actually be made. So, this lesson is optional. It is a good idea to get used to using the import function, anyway.

1. Log in as the root user.
2. Open a Terminal window
3. Start the DHCP manager with the command:  
`/usr/sadm/admin/bin/dhcpmgr &`
4. Left click on Service in the Menu bar.
5. Left click on Import Data.



6. Type in the location **/myexportfile**
7. Make sure to select the  "Overwrite existing data" checkbox
8. Left click on the **>** button.
9. Left click on the Finish button.
10. Left click on **S**ervice.
11. Left click on **S**top.
12. Left click on **S**ervice.
13. Left click on **S**tart.
14. Left click on **F**ile and then on **E**xit to exit the DHCP server.

## Command Line Tools to Manage a DHCP Server

The DHCP manager is an easy tool to use when a graphical environment is available. Unfortunately, there are plenty of situations where a company does not install the CDE or OpenWindows environment on a server. In this case, the only way to manage a DHCP server is with command line tools.

DHCP commands are also used in scripts set up by the system administrator. For example, an administrator might create scripts that check the availability of IP addresses and then dynamically add or remove addresses from a DHCP server. By executing DHCP command line utilities, a program or script, can set up a very powerful, custom made DHCP server, something that can not be done with a GUI interface like the DHCP Manager.

There are three main command line tools used to manage a DHCP server:

- The **dhcpconfig** command is used to start, stop, enable and disable the DHCP server. It can also be used to export the DHCP server's data to a DHCP configuration file. In addition, this command can completely remove DHCP from a Solaris 9 server.

- The **dhtadm** command is used to configure macros and options. This command is very cryptic and difficult to learn.
- The **pntadm** command is used for day to day maintenance. It is mostly used to set up networks and IP addresses with DHCP.

### The **dhcpconfig** Command

The **dhcpconfig** command is the primary command used to set up a DHCP server. This tool basically performs the same configuration and policy setup functions as the setup wizards in the DHCP Manager.

The following options are supported by the **dhcpconfig** command:

#### **dhcpconfig**

- U Un-configures a DHCP server. The following secondary options are supported:
  - f Do not display the “are you sure?” confirmation message
  - h Removes all the host names
  - x Removes the network tables and the **dhcptab** file
- D Specifies various parameters:
  - a IP addresses of one or more DNS servers, separated by the comma ( , ) symbol
  - h Where to place host data
  - l Lease time in seconds
  - n Non-negotiable leases for DHCP information
  - u Data ignored by **dhcpconfig** that is placed in the DHCP database
  - y NIS+ or DNS domain; only works if DNS or NIS+ is specified
- I Imports DHCP server configuration information from an export file.
- X Exports DHCP server configuration information to an export file.
  - a List of network and addresses to export; ALL is a recognized keyword
  - m List of macros to export; ALL is a recognized keyword
  - o List of options to export; ALL is a recognized keyword
  - x Clears out the DHCP server after the export file is created

### The **dhtadm** command

The **dhtadm** command is used to configure macros and symbols in the DHCP table.

#### **dhtadm**

- A Adds a macro or symbol to the DHCP table.
  - d Defines a macro or symbol
  - m Designates a macro name, must be used with **-d**
  - s Designates a symbol name, must be used with **-d**
- B Batch process of **dhtadm** commands from a text file or STDIN.
  - v Shows verbose output as the batch file is being read
- C Creates the **dhcptab** table.

- D Deletes the **dhcptab** table or a **dhcptab** table entry.
  - m Designates a macro name
  - s Designates a symbol name
- M Modifies existing macros or symbols.
  - d Defines a macro or symbol
  - e Edits a macro or symbol
  - m Designates a macro
  - n Renames a macro or symbol
- P Prints the **dhcptab** table.
  - r Specifies a different resource than the one specified in the `/etc/init/dhcpsvc.conf` variable **RESOURCE**.
- R Removes the **dhcptab** table

## The **pntadm** Command

The **pntadm** command works with networks and IP addresses.

- pntadm** This command manages the DHCP network tables. Common functions include showing IP addresses used, assigning new IP address pools and removing IP addresses.
- A Adds a hostname and IP address.
    - c Comment
    - e Expiration date of lease, in MM/DD/YYYY format (rarely used)
    - f Flag value, can use keywords or numbers
 

|           |    |                                                                                          |
|-----------|----|------------------------------------------------------------------------------------------|
| DYNAMIC   | 00 | Server's default assignment                                                              |
| PERMANENT | 01 | Lease never ends (good for routers)                                                      |
| MANUAL    | 02 | Administrator specifies this                                                             |
| UNUSABLE  | 04 | Not a valid IP address (blocks IP address from being used; servers usually have this IP) |
| BOOTP     | 08 | Used with diskless clients and JumpStart installations                                   |
    - h Hostname
    - i Client identifier (hex)
    - m Macro assigned to this client
    - s DHCP server or IP name
  - B Batch process of **dhtadm** commands from a text file or STDIN.
    - v Shows verbose output as the batch file is being read
  - D Deletes the hostname or IP address.
    - y Deletes the hostname from the name service
  - L Lists the DHCP table of network addresses.
  - M Modifies the hostname or IP address.
    - c Comment
    - e Expiration date of lease, in MM/DD/YYYY format (rarely used)
    - f Flag value, can use keywords or numbers
 

|           |    |                                     |
|-----------|----|-------------------------------------|
| DYNAMIC   | 00 | Server's default assignment         |
| PERMANENT | 01 | Lease never ends (good for routers) |

- |  |          |    |                                                                                          |
|--|----------|----|------------------------------------------------------------------------------------------|
|  | MANUAL   | 02 | Administrator specifies this                                                             |
|  | UNUSABLE | 04 | Not a valid IP address (blocks IP address from being used; servers usually have this IP) |
|  | BOOTP    | 08 | Used with diskless clients and JumpStart installations                                   |
- h** Hostname
  - i** Client identifier (Hex)
  - m** Macro assigned to this client
  - s** DHCP server or IP name
- p** Sets the location of the path to the DHCP database source. This is used instead of the PATH variable specified in the `/etc/init/dhcpsvc.conf` file.
  - P** Prints the IP addresses assigned and used in a `dhcptab` table.
  - R** Removes the network from DHCP.
  - r** Specifies the source for the data. This is the same as the **RESOURCE** variable used in the `/etc/init/dhcpsvc.conf` file.
  - u** Sends the data to a public module and tells `pntadm` to ignore the data.

## Lesson 31.7 Using the `pntadm` Command

In this lesson readers will make a fictitious network inside the DHCP server. Several IP addresses will be added to the fictitious network. You can then play around with the fictitious network before deleting it from the server.

- Log in as the root user.
- Open a Terminal window.
- Type the command `pntadm -P 192.168.0.0` (or any other valid network number).  
*The command `pntadm -P <network_number>` shows all the IP addresses that have been assigned to that network.*
- Type the command `pntadm -C 100.100.0.0`  
*This command creates a fictitious network segment with the IP address 100.100.0.0.*
- Type the command `pntadm -r SUNWfiles -p /var/dhcp -A 100.100.0.101 100.100.0.102`  
*This command uses `-r` to override the variable `RESOURCE=` from `dhcpsvc.conf` file. It then uses `-p` to display the `dhcptab` table, and `-A` to add two clients to the DHCP server.*
- Type the command `pntadm -P 100.100.0.0`  
*This shows all the IP addresses that have been assigned to that network.*
- Type the command `pntadm -A 100.100.0.102 -c "This is good" 100.100.0.0`  
*The command `pntadm -A` is somewhat tricky. Immediately after the 100.100.0.102 (new IP address) is one of the subarguments to `-A` (which in this case is `-c "comment"`). The last argument is the network that this IP address will be applied to, which in this case is 100.100.0.0.*
- Type the command `pntadm -P 100.100.0.0`  
*Note the change in this output from before.*
- Type the command `pntadm -M 100.100.0.1 -c "This is bad" 100.100.0.0`  
*The command `pntadm -M` modifies the comment on the 100.100.0.1 IP address to say "This is bad" instead of "This is good."*
- Type the command `pntadm -P 100.100.0.0`  
*Note the change in this output from before.*

## Lesson 31.8 Starting and Stopping the DHCP Server From the Command Line

This is a rather uneventful lesson. The DHCP server starts and stops from the command line. This is useful if the DHCP server is modified but for some unknown reasons the new changes do not seem to take effect.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command `/etc/init.d/dhcp stop`
4. Type the command `/etc/init.d/dhcp start`

### Clearing a DHCP Server

Sometimes, it is useful to "clear" a DHCP server. When this is done, the server's configuration information is removed. The server will not be operational until a configuration is restored. Usually, the new configuration includes changes from the previous configuration.

When a DHCP server is cleared, its DHCP clients still keep their current network settings, until their leases expire. After a client's lease expires, the client will attempt to contact the previous server. If the DHCP server that initially set up the client can not be found, the client will try to contact other DHCP servers for DHCP information. Eventually, if no DHCP server can be found, the network will come to a grinding halt. Of course, you should avoid this situation.

When a DHCP server is cleared, the DHCP software is not removed from the system. Remember, though, that as long as the server has no configuration information, it will not try to answer DHCP requests. When the configuration information is restored, the server will be operational again.

## Lesson 31.9 Using `dhcpcfg` to Clear and Restore a DHCP Server

In this lesson, you will save the configuration of the DHCP server to an export file. Then, you will stop the DHCP server and clear its configuration. Finally, you will stop the DHCP server, restart it, and restore its configuration from the saved export file.

**Note:** This lesson requires that you have a running DHCP server. If this is not the case, follow Lesson 31.2 to create a simple DHCP server. Then continue with this lesson.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command  
`dhcpcfg -X /myexportfile2 -a ALL -m ALL -o ALL`  
*This command saves the DHCP server configuration to the export file /myexportfile2.  
The keyword ALL is case sensitive, so make sure that you type it in all caps.*
4. Type the command `dhcpcfg -U -h -x`  
*The options -U -h -x are used to remove the hostnames from the DHCP server, from the DHCP network tables, and from the dhcptab file. At this point in time the DHCP server has been removed from the system. The software still exists, but the server has no configuration.*
5. Type the command `dhcpcfg -D -r SUNWfiles -p /var/DHCP`  
*This command adds configuration information to the DHCP server.*
6. Type the command `dhcpcfg -I /myexportfile2`  
*This command restores the original DHCP server to its original condition by reading in the configuration information saved in /myexportfile2..*

## Lesson 31.10 Using Command Line Utilities with a Solaris 9 DHCP Client

This lesson introduces readers to various Solaris 9 DHCP command line utilities that provide information about your DHCP setup. These utilities must be run on a DHCP client. Remember, a DHCP server can not also be a DHCP client. If you only have one Solaris 9 test system to work with, you can create a DHCP server on another system (such as Linux, FreeBSD or Windows 2000) and then configure Solaris 9 to be a DHCP client.

1. Log in as the root user.
2. Open a Terminal window.
3. Type the command `ls /etc/dhcp*`  
*This command shows all files that start with the first four letters **dhcp**. In this case there needs to be a file named **/etc/dhcp.network\_card** for each Ethernet card used with DHCP. If the file **/etc/dhcp.network\_card** does not exist, then DHCP is not running on this client.*
4. Type the command `ifconfig -a`  
*This command shows the current configuration of the network card on the current workstation.*
5. Type the command `cat /etc/dhcp/if.dhc`  
*This shows the current DHCP configuration of the network card.*
6. Type the command `cat /etc/default/dhcpagent`  
*This shows the tunable parameters of DHCP. Do not change these by hand unless there is a very strong reason for doing so!*
7. Type the command `dhcplinfo -c 3`  
*The identifier **3** is for a router.*
8. Type the command `dhcplinfo -c Ip`  
*The symbol **Ip** represents the IP address assigned to the system.*
9. Type the command `dhcplinfo -c 72`  
*The identifier **72** is used to identify a World Wide Web server. This might not be a DHCP parameter that has been given by the DHCP server*

## Lesson 31.11 Working with the dhtadm Command

In this lesson, you will use the `dhtadm` command to change the `dhcptab` table from the command line. You will then print out the `dhcptab` table, add an entry to it, and then delete the entry from this table.

1. Login as the root user.
2. Open a Terminal window
3. Type the command `/usr/sbin/dhtadm -P`  
*This command prints the **dhcptab** table. This will look something like:*

| Name        | Type  | Value                                                               |
|-------------|-------|---------------------------------------------------------------------|
| 192.168.0.0 | Macro | Subnet=255.255.255.0:Router=192.168.0.1<br>:Broadcst=192.168.0.255: |
| sun100      | Macro | :Include=Locale:Timeserv=192.168.0.4:<br>LeaseTim=86400:LeaseNeg:   |
| Locale      | Macro | :UTCoffst=-25200:                                                   |

*This command shows the following pieces of information about the DHCP table:*

*Name* Name of the symbol record.  
*Type* Type of record being displayed.  
*Value* Variables used with this record. In the above example, some of these include the DHCP subnet value, the router and the broadcast address. To obtain further information on DHCP tables, type the command `man -s4 dhcptab`. This is the man page on the DHCP table.

4. Type the command `dhtadm -A -m newdns -d `:DNSserv=192.168.0.5:``  
This command adds a new macro to the `dhcptab` table with the name `newdns` that contains an entry for a DNS server.
5. Type the command `dhtadm -P`  
This command prints out the `dhcptab` table. Notice that this time there is a new macro with the name `newdns` that contains the value `DNSserv=192.168.0.4`, as shown below:

|  | Name                     | Type  | Value                                                                              |
|--|--------------------------|-------|------------------------------------------------------------------------------------|
|  | <code>newdns</code>      | Macro | <code>:DNSserv=192.168.0.4:</code>                                                 |
|  | <code>192.168.0.0</code> | Macro | <code>:Subnet=255.255.255.0:Router=192.168.0.1<br/>:Broadcst=192.168.0.255:</code> |
|  | <code>sun100</code>      | Macro | <code>:Include=Locale:Timeserv=192.168.0.4:<br/>LeaseTim=86400:LeaseNeg:</code>    |
|  | <code>Locale</code>      | Macro | <code>:UTCoffst=-25200:</code>                                                     |

6. Type the command `dhtadm -M -m newdns -d `:DNSserv=192.168.0.7:``  
This command modifies the `newdns` macro. The variable `DNSserv` was changed from `192.168.0.4` to `192.168.0.7`.
7. Type the command `dhtadm -P`  
This command prints the `dhcptab` table. The table should look the same as above, except that the macro named `newdns` has been modified.
8. Type the command `dhtadm -D -m newdns`  
This command deletes the `newdns` macro from the `dhcptab` table.
9. Type the command `dhtadm -P`  
This command prints the `dhcptab` table. Notice that the `newdns` macro is gone?

## Client Side DHCP Files

These files reside on the Solaris 9 DHCP client. They are created automatically when a Solaris 9 workstation becomes a DHCP client. These files contain the DHCP settings that the client has obtained from the DHCP server. If you believe that for some reason the DHCP settings are incorrect, examine your DHCP server. Do not automatically assume that the DHCP client is the problem.

### `/etc/dhcp/if.dhc`

This file shows the configuration of a DHCP interface on the client.

### `/etc/default/dhcpagent`

This file shows the tunable parameters for the DHCP protocol. Unless there is a severe problem with DHCP, these values should not be changed.

## Client and Server Side DHCP Commands and Daemons

These commands and daemons reside on both the Solaris 9 DHCP client and server.

**in.dhcpd** This is the actual DHCP daemon. If clients can not receive DHCP information, type the command `ps -ef | grep in.dhcpd` to see if this daemon is running.

If the daemon is not running, use the DHCP Manager to start the DHCP server.

**dhcpagent** This resides on the DHCP client. When Solaris 9 starts, the `/sbin/ifconfig` command looks for a file named `/etc/dhcp.<network_adapter>`. If this file exists, the `dhcpagent` contacts the DHCP server for the network configuration. The `dhcpagent` also obtains a lease

time from the DHCP server. If the lease expires, the **dhcpcagent** obtains an extension to the lease or new network parameters. If a new lease can not be obtained, the network interface is disabled.

**dhcpcinfo** The **dhcpcinfo** command is used to show the DHCP values that were given to the DHCP client by the DHCP server.

### **The /etc/init.d/dhcp Script**

The **/etc/init.d/dhcp** command is called by the run control scripts when the Solaris 9 operating system starts. The command **/etc/init.d/dhcp start** begins the **in.dhpcd** daemon.

### **Key Points to Remember**

DHCP is a very handy tool to have when dealing with a large scale network. The DHCP server can be set up to automatically assign IP addresses to a wide variety of networked devices. The Solaris DHCP server has a very nice GUI interface that lets the system administrator easily set up the DHCP server. Make sure to also practice using the DHCP command line tools, so that if you come across a Solaris server that does not have a GUI interface, you will still be able to save and modify the DHCP server configuration.

# Index

---

- .
  - .csrc**, 7-5, 8-4
  - .dt**, 3-3, 3-33, 3-36, 3-37, 7-20, 7-30, 7-32
  - .dtprofile**, 3-3, 3-35, 3-36, 3-37, 7-20
  - .exrc** File, 4-1, 4-7
  - .korn**, 8-4
  - .profile**, 2-37, 3-36, 5-7, 7-5, 7-6, 7-11, 7-20, 7-23, 7-31, 7-34, 8-4, 12-2, 12-35
  - .Z** file extension, 18-15
  - .z** files, 18-15
- /
- /bin**, 7-2, 7-5, 7-6, 7-7, 7-21, 7-25, 8-1, 8-2, 8-3, 8-7, 8-8, 12-3, 12-5, 12-7, 12-8, 12-9, 12-10, 12-11, 12-12, 12-13, 12-14, 12-15, 12-16, 12-17, 12-19, 12-20, 12-21, 12-22, 12-23, 12-24, 12-25, 12-26, 12-27, 12-28, 12-29, 12-30, 12-31, 12-32, 12-33, 12-35, 17-12, 30-22, 30-55
  - /bin/csh**. See C shell, See C shell, See C shell
  - /bin/ksh**. See Korn shell, See Korn shell
  - /bin/sh**. See Bourne shell
  - /cdrom/cdrom0/s0/Solaris\_9/Tools**, 21-22
  - /cdrom/cdrom0/Solaris\_9/Tools**, 21-23
  - /dev**, 2-8
  - /dev** directory, 8-1, 8-2, 13-8, 13-10, 13-11
  - /dev/cua**. See modem
  - /dev/cus**, 8-2
  - /dev/dsk**, 7-32, 7-33, 7-34, 7-35, 7-36, 8-1, 8-2, 8-10, 13-9, 13-10, 13-11, 14-2, 14-4, 14-6, 14-7, 14-9, 15-9, 15-10, 18-8, 18-18, 20-5, 21-3, 21-11, 25-2, 25-3, 25-9, 25-10, 26-13, 26-14, 26-15, 26-21
  - /dev/fbs**, 8-2
  - /dev/fd**, 7-32, 7-33, 8-2, 14-2, 15-2
  - /dev/printers/0**, 17-7
  - /dev/pts**, 8-2
  - /dev/rdisk**, 2-36, 7-32, 7-33, 7-36, 8-2, 11-2, 14-3, 14-9, 14-10, 15-5, 15-11, 15-12, 18-6, 18-8, 18-9, 18-10, 18-16, 18-18, 21-17, 21-18, 25-3, 26-20, 26-21
  - /dev/rmt**, 2-8, 8-2, 18-5, 18-6, 18-7
  - /dev/rmt/0**, 18-7
  - /dev/term**, 8-2
  - /devices**, 2-8
  - /devices** directory, 2-8, 2-36, 8-1, 8-2, 8-6, 13-10, 13-11
  - /etc** directory, 4-3, 4-4, 4-5, 4-13, 5-4, 7-2, 8-1, 8-3, 12-13, 12-14, 15-9, 16-10, 16-11, 18-11, 18-12, 23-2, 23-3, 23-4, 23-5, 23-13, 23-17, 23-18, 23-19, 29-3, 30-61
  - /etc/.rootkey**, 23-22
  - /etc/acct**, 8-3, 28-8
  - /etc/acct/holiday**, 28-6, 28-12
  - /etc/acct/holidays**, 28-4, 28-8
  - /etc/apache**, 29-19
  - /etc/coreadm**, 25-4
  - /etc/cron.d**, 8-3, 11-3, 11-5
  - /etc/default/dhcpagent**, 31-38
  - /etc/defaultdomain**, 16-10, 23-9, 23-19, 29-14, 30-9
  - /etc/defaultrouter**, 16-10, 16-12, 29-14, 29-17, 29-20
  - /etc/dfs/dfstab**, 14-12, 14-13, 14-14, 21-26, 21-28, 21-32
  - /etc/dfs/fstypes**, 14-10, 14-11
  - /etc/dhcp/if.dhc**, 31-38
  - /etc/dumpadm.conf**, 25-9
  - /etc/dumpdates**, 18-6
  - /etc/format.dat**, 13-13, 13-14
  - /etc/group**, 7-2, 7-3, 7-5, 7-6, 7-7, 7-15, 7-16, 7-20, 7-22, 7-23, 7-29, 19-16, 23-3, 23-4
  - /etc/groups**, 4-5
  - /etc/holidays**, 28-13
  - /etc/hosts**, 2-4, 14-13, 16-7, 16-8, 16-9, 16-12, 16-13, 21-29, 21-30, 21-31, 23-1, 23-3, 23-4, 23-9, 23-12, 23-18, 23-19, 23-20, 27-6, 29-1, 29-3, 29-6, 29-14, 29-17, 29-20, 31-13
  - /etc/inet/inetd.conf**, 16-14, 16-15
  - /etc/inet/ipnodes**, 27-6
  - /etc/inetd.conf**, 27-6
  - /etc/init.d/acct**, 28-3, 28-18
  - /etc/init.d/rpc**, 23-22
  - /etc/init.d/syslog**, 24-1
  - /etc/inittab**, 6-4, 6-5, 6-6, 6-7, 6-9
  - /etc/initttab**, 6-4
  - /etc/lp**, 8-3, 17-4, 17-8, 17-9, 17-10
  - /etc/lp/printers/<printer\_name>**, 17-4
  - /etc/lvm/mddb.cf**, 26-15
  - /etc/mnttab**, 14-2, 14-9, 14-10
  - /etc/named.conf**, 29-7, 29-8, 29-14
  - /etc/netmasks**, 29-14
  - /etc/nodename**, 16-10
  - /etc/nsswitch.conf**, 23-8, 23-12, 23-18, 29-2, 29-6, 29-14
  - /etc/nsswitch.files**, 23-8
  - /etc/nsswitch.nisplus**, 23-18
  - /etc/passwd**, 4-5, 4-17, 7-2, 19-15, 23-6, 23-22, 29-4
  - /etc/path\_to\_inst**, 13-11
  - /etc/printers.conf**, 17-12
  - /etc/profile**, 5-7
  - /etc/project**, 19-16, 19-17
  - /etc/rc0.d**, 12-3
  - /etc/rc0.d/K22acct**, 28-3, 28-17
  - /etc/rc1.d**, 12-3
  - /etc/rc2.d**, 12-3
  - /etc/rc2.d/S22acct**, 28-3, 28-17

/etc/rc3.d, 12-3  
 /etc/release, 4-9  
 /etc/resolv.conf, 29-2, 29-14, 29-18  
 /etc/rpcinfo, 16-15  
 /etc/security/auth\_attr, 22-4, 22-6, 22-7, 22-16  
 /etc/security/exec\_attr, 22-5, 22-6, 22-16  
 /etc/security/prof\_attr, 22-4, 22-6, 22-7  
 /etc/services, 16-14, 16-15, 23-5  
 /etc/shadow, 4-5, 7-2, 7-4, 7-6, 7-8, 7-23, 7-26, 7-29, 23-6  
 /etc/skel, 7-20, 8-4  
 /etc/syslog.conf, 24-1, 24-2, **24-4**  
 /etc/user\_attr, 22-6, 22-16  
 /etc/vfstab, 7-32, 7-33, 7-36, 14-1, 14-2, 14-7, 14-8, 14-9, 14-10, **14-15**, 15-11, 21-11, 25-3, 26-21, 26-26, 26-30, 26-34  
 /export/home, 21-4  
 /JumpStart directory, 21-16  
 /mnt Directory, **8-4**  
 /opt/Acrobat5/bin/acroread, 3-17  
 /s0/Solaris\_9/Tools, 21-22  
 /sbin Directory, **8-5**  
 /sbin/ifconfig, 16-10, 31-38  
 /tmp, 4-3, 7-8, 7-9, 7-32, **8-5**, 9-6, 14-2, 15-2, 17-8  
 /tmp Directory, **8-5**  
 /usr Directory, **8-5**  
 /usr/bin, 3-24, 4-4, 5-4, 8-3, **8-5**, 8-7, 8-8, 9-9, 9-10, 12-8, 12-33, 12-34, 17-3, 25-7, 28-3, 30-55  
 /usr/bin/admintool, 3-24, 25-7  
 /usr/ccs/bin/make, 23-5, 23-10  
 /usr/kernel/drv/unix.<x>, 25-8  
 /usr/kernel/drv/vmcore.<x>, 25-8  
 /usr/lib/acct, 28-4  
 /usr/lib/acct/acctmerg, 28-19  
 /usr/lib/acct/monacct, 28-16  
 /usr/lib/acct/prdaily, 28-9, 28-16  
 /usr/lib/acct/runacct, 28-9, 28-13, 28-16  
 /usr/lib/lp/model/standard, 17-4  
 /usr/lib/lp/netstandard, 17-4  
 /usr/lib/lp/postscript, 17-4  
 /usr/lib/netsvc/yp/ypstart, 23-10  
 /usr/lib/nis/nisclient, 23-22  
 /usr/lib/nis/nisping, 23-19  
 /usr/lib/nis/nisserver, 23-18, 23-19  
 /usr/sadm/admin/bin/dhcpmgr, 31-6  
 /usr/sbin/dhccpconfig, 31-6  
 /usr/sbin/ifconfig, 16-10, 31-23  
 /usr/sbin/in.named, 29-16  
 /usr/sbin/install.d/pfinstall, 21-18  
 /usr/sbin/metainit, 26-34  
 /usr/sbin/metattach, 26-34  
 /usr/sbin/mkfile, 7-33, 7-36  
 /usr/sbin/quotaoff, 7-33  
 /usr/sbin/quotaon, 7-33

/usr/sbin/startconsole, 30-14  
 /usr/sbin/ypinit, 23-10  
 /usr/share/lib/terminfo, 17-3, 17-4, 17-5  
 /var/adm, 24-1  
 /var/adm/acct/fiscal, 28-6, 28-15  
 /var/adm/acct/nite, 28-13  
 /var/adm/acct/nite/, 28-6  
 /var/adm/acct/sum, 28-6  
 /var/adm/acct/sum/tacct, 28-6  
 /var/adm/fee, 28-6, 28-7  
 /var/adm/messages, 19-13, 24-1  
 /var/adm/pacct, 28-4, 28-18  
 /var/adm/utmpx, 28-6  
 /var/adm/wtmpx, 28-6, 28-11  
 /var/crash/<hostname> directory, 25-6  
 /var/log/syslog, 19-13  
 /var/named, 29-8, 29-9, 29-14  
 /var/named/domain-info, 29-15  
 /var/named/inverse-domain-info, 29-10, 29-15  
 /var/named/loopback-domain-info, 29-10  
 /var/named/named.root, 29-8, 29-16  
 /var/nis/data, 23-20  
 /var/sadm/install/logs/, 2-35, 21-21  
 /var/sadm/patch, 10-2, 10-3, 10-4, 10-5  
 /var/sadm/pkg, 9-3  
 /var/spool/cron/crontabs, 11-1, 11-2  
 /var/spool/lp, 17-3  
 /var/spool/lp/system/pstatus, 17-3  
 /var/spool/pkg, 9-6, 22-11  
 /var/yp, 23-3, 23-5, 23-12  
 /var/yp/Makefile, 23-5

“

“/var/named, 29-7

## 1

13W3, 1-3, 1-4, 1-14  
 13W3 Female to HD-15 Male, 1-3  
 13W3 MALE to SVGA FEMALE, 1-4  
 13W3 to SVGA adapter, 1-3  
 13W3 video port, 1-3

## 9

9\_Recommended, 10-4, 10-8  
**9\_Recommended.zip**, 10-4

## A

ABI, 9-2  
 accept/reject, 17-11  
 Access Control Information. *See* ACI  
 acctwtmp, 28-11  
 ACI, 30-2  
**ACK.** *See* acknowledgement packet

**active0624**, 28-6  
 actuator arm, 13-5  
 adb, **25-6**  
**adb** command, 25-6  
*adb debugger*, 25-7  
 Add Local Printer window, 17-7  
**add\_install\_client**, 21-23  
**add\_install\_server**, 21-25  
**add\_install\_server** script, 21-23  
**add\_to\_install\_server**, 21-5, 21-23, 21-26  
**adm**, 28-4  
*Administration Server*, 30-13  
*administrative shell*, 22-4  
 Admintool, 3-1, 3-2, 3-7, **3-24**, 3-31, 3-34, **7-1**, 7-2, **7-3**, **7-5**, 7-7, **7-16**, 7-17, **7-31**, 9-1, 9-7, **9-9**, 11-6, **17-1**, 17-2, **17-6**, **19-4**, 19-17, 25-7  
 Adobe Acrobat, 3-1, 3-16, 3-17, **5-1**, 5-2, 30-5  
 Adobe Acrobat 5.0.5 reader, 3-17  
 Adobe Acrobat Reader, 30-4  
**anon**, 14-11  
 AnswerBook2, 3-16, 3-31  
 Anycast Address, 27-1, 27-3  
 AOL, 3-27  
 Apache Web server, 2-2, 28-14, 29-19  
**Apache Web Server**, **9-7**, **9-9**, 20-3, **29-1**, 29-19  
 Application Binary Interface. See ABI  
**Application Layer**, **16-3**, **16-7**, **16-9**  
 Application Manager, 3-31, 3-36  
**apropos**, 5-4, 5-6  
**at** command, 8-3, **11-1**, **11-4**, **11-5**, **11-6**  
**at.allow**, 11-5  
**at.deny**, 11-5  
 atomic print command, 17-11  
**Audit Control**, 22-2  
**Audit Review**, 22-2  
 auth\_attr database, 22-7  
**authname**, 22-7  
 authorization, 22-4  
*authorizations*, 22-6  
**auths**, 22-5  
**auto-boot**, 1-12, 1-16, 1-17, **1-18**, 1-20, 1-21, 2-5, 2-6, 6-2, **6-3**, 21-6  
**auto-boot?**, 1-12, 1-16, 1-17, **1-18**, 1-20, 1-21, 2-5, 2-6, 6-2, **6-3**, 21-6  
**Avontra**, i

## B

Backdrop, 3-30, 3-31  
 background image, 3-29, 3-30  
 backplane board, 1-13  
 Backtrack, 3-15  
 backup, 18-2, 18-3, 18-4  
 Backup Log Files, 19-13  
 backup schedule, 18-3  
**backup\_media**, 21-11  
**banner**, 1-16, 1-20, 1-21, 6-1, 6-2, **16-1**, **16-12**, 16-13, **17-9**, 21-31, 21-32, 30-19

**banner** command. See banner, See banner  
**beginning\_script\_file**, 21-22  
 Beta Refresh version of, 1-5  
 binary  
   conversion chart, 27-11  
 Binary License Program, 1-5  
 binary to decimal, 3-26, 27-10, 27-11  
 binary to decimal conversions, 3-26  
 BIOS, 1-10, 1-11, 1-16, **1-18**  
 block, 28-1  
 block device, 8-6  
*block device* file, 8-6, 26-2  
 Block Device files, 8-6  
 block mode, 13-2  
**boot**, 1-17, **1-18**  
**boot cdrom**, 1-18, 2-1, **2-2**, 2-6, 2-7, 6-2, *18-18*, 21-1, 21-4, 21-6, 21-20  
**boot cdrom - install**, 21-7  
**boot command**, 18-18, 25-10  
**boot disk**, 1-18, 3-11, 13-2, **13-8**, **21-1**, 21-2  
**boot net - install**, 21-6  
**boot network**, 1-18, **21-5**, 21-32  
 boot partition, 21-25  
 boot process, 6-1  
 Boot server, 21-6  
**Boot Server**, 21-5, 21-22  
**boot\_device**, 21-12  
*bootblk*, 6-3, 15-2, 15-3, **15-4**, 21-28  
**boot-device**, 21-2, 21-21  
**bootp**, 21-5  
 BOOTP packet, 31-1, 31-2  
 BOOTP packets, 31-6  
 BOOTP relay agent, 31-1  
**Bootparams**, 23-4  
 Bourne shell, 1-5, 3-36, **7-3**, **7-6**, **7-21**, **7-23**, **7-25**, 12-2, 12-3, **12-8**, 12-9, 12-12, *12-14*, *12-15*, *12-16*, 12-18, **12-25**, **12-32**, **12-34**, **12-35**, 12-36, **17-4**, 21-9, 21-18, **21-22**, **21-24**, **28-2**, 30-6, 30-22, 30-51, 30-54  
 Bourne Shell, 7-5, **7-7**, **7-22**, **7-25**, 12-1, *12-3*, 12-4, **12-8**, 12-9, *12-14*, **12-25**, **12-32**, 12-36, **22-16**, 30-58  
**break** command, 12-31  
 broadcast address, 31-1, 31-2  
 Browser, **2-21**, 3-27, 30-1, 30-52  
*Bundled Software Package*, 9-2

## C

C shell, 3-36, **7-3**, 7-7, **7-20**, **7-23**, **7-25**, *12-2*, *12-14*, 20-10, 22-1  
 C Shell, 7-5, **7-22**, 12-1  
 Calculator, 3-26  
 Calculator Utility, **3-26**  
 Calendar, 3-2, 3-4  
**case** command, 12-22, **12-23**, **12-24**, **12-25**, **12-36**

**cat** command, 3-29, 4-5, **4-6**, 4-8, 4-16, **7-15**, **7-16**,  
 12-13, 12-26, 21-21, **23-5**, **23-9**, **23-20**, 23-22  
**CatchKey( )** function, 12-36  
**catman -w**, **5-4**, **5-6**, 5-7  
 CCITT, 30-3  
**cd** command, 4-2, 4-3, **4-4**, **7-13**, 8-2, 8-8  
 CDE, 2-5, 2-10, 2-11, 2-28, 2-38, **2-39**, **3-1**, 3-2, **3-3**, 3-  
 5, 3-6, 3-8, 3-9, 3-10, 3-11, **3-12**, 3-13, 3-14, 3-15, 3-  
 19, 3-30, 3-31, **3-32**, **3-33**, 3-35, 3-36, 3-37, 4-1, 4-3,  
**4-4**, 4-6, 6-7, **7-1**, 7-11, **7-20**, **7-30**, **7-31**, **7-32**, 7-33,  
**8-5**, **9-5**, **12-34**, **19-1**, 20-4, 20-8, **21-12**, **21-15**,  
 26-1, 31-32  
 CDE desktop, 2-38, 3-1, **3-3**, 3-36, 3-37  
 Character Device files, 8-6  
 Character mode, 8-6  
**chargefee**, **28-13**  
**chargfee**, **28-2**  
**CHARSETDIR**, **17-8**  
**check** command, 21-7, 21-8, 21-17  
**check** program, 21-23  
**Check script**, **21-5**  
**chmod**, **7-9**, 7-10  
**chmod 777**, **6-8**  
**chmod** command, 7-10, **7-13**, 12-3, 12-5, 12-6, 12-10  
 Cisco, ii  
**ckpacct**, **28-7**  
**Class A address**, **16-8**  
**Class B address**, **16-8**  
 class B network, 16-8  
**Class C address**, **16-8**  
**Class D address**, **16-8**, **16-10**  
**Class E address**, **16-8**  
**clear** command, **26-14**  
 clock board, 1-13  
**Cluster**, **21-12**  
**cms**, **28-7**  
 Color Server, 3-36  
 Color Style Manager, 3-29, 3-31  
**Command Line Login**, **2-37**, **2-39**, 3-9  
 Command Line mode, 3-9  
 Command Line session, 3-9, 3-11  
 Command Line Session, 3-2, 3-9, 3-11  
 Command Line Tools for Floppy Disks, **3-21**  
 Command Mode, 4-5  
 Common Desktop Environment, xvi, 2-10, 3-1, 3-15, 4-1,  
**8-5**, **9-5**, 20-4  
 Common Open Software Environment. *See* COSE  
**compress**, **18-12**, 18-15  
**compress** utility, 18-15  
 concatenated volume, 26-28  
 Concatenated Volumes, 26-4, **26-32**  
 Configuration server, 21-6  
**Configuration Server**, **21-4**, 21-22  
 console login prompt, 2-37, **2-39**, **2-40**  
 Consultative Committee on International Telegraphy and  
 Telephones. *See* CCITT  
**continue** command, 12-31, 12-32

control statements, 12-18, 12-29  
**Copyright**, **1-25**, 2-8, 2-35, **10-5**  
**core**, **25-1**  
*core file*, 25-4  
*core file*, 25-7  
**core** files, 25-6  
**coreadm**, **25-1**  
     Patterns Used, **25-5**  
**coreadm** command, 25-4, 25-6  
**coreadm** Command, **25-5**  
 COSE, 3-1  
 CPU Disk Activity, 3-4  
 Creating a Custom Made Icon, **3-33**  
**cron** command, 28-5, **28-7**  
**cron** daemon, 22-2  
**cron** file, 28-7, **28-16**, 28-17  
 cron job, 11-1, 28-11  
**Cron Management**, **22-2**  
 cron.allow, **11-3**, **11-5**  
 cron.deny, **11-3**, **11-5**  
**crontab**, **11-1**, **11-2**, **11-3**, **11-4**, **11-5**, 11-8, **19-4**,  
**28-7**, **28-8**, 28-17  
**crontab** command fields, 11-2  
**crontab** file, 28-17  
**CTL + BREAK**, **2-5**  
 CTL + Break keys, 21-6  
**Custom Jumpstart**, **2-2**, 2-9, **21-4**  
 Custom Jumpstart Installation, **21-4**  
 Custom JumpStart installation, 21-2  
 cylinder, 7-30, 13-2, **13-6**, **13-8**, 15-2, 15-3, **15-4**, 15-5,  
 15-6, 15-11, **15-13**  
 cylinder group, 7-30, **15-3**, **15-4**, 15-5, 15-6, 15-11, **15-13**  
**Cylinder Group Block**, **15-3**, 15-4, 15-6  
 cylinders, 13-6, **13-8**, **13-13**, **13-14**, **13-15**, 15-2, **15-3**,  
**15-4**, 15-5, 15-6, **15-13**

## D

**daemon** account, **19-15**  
 Daily Command Summary, 28-14  
 Daily Report, 28-10, 28-11, 28-14  
     CPU time, 28-11  
     prime, 28-11  
     user's activity, 28-10  
 DAP, 30-3  
 data blocks, 7-30, 8-9, **15-3**, **15-4**, 15-5, 15-6, 15-7, 15-8,  
 15-11, 18-17, 26-8, 28-1  
**Data Blocks**, **15-3**, 15-4, 15-7  
**Data Link layer**, **16-5**  
 date command, 28-11, 28-17  
**daycms**, **28-6**  
**dd** command, 18-16  
 Debugging, **12-32**  
 decimal to binary, 3-26  
 default font size, 3-29, 3-30  
*default gateway*, 2-15

- default printer, 17-12
- default router, 31-1
- Default Router, 2-4, 29-13
- default.csrc, 8-4**
- default.korn, 8-4**
- default.profile, 8-4**
- denial of service attacks, 27-7
- DES authentication, 23-14
- desktop color theme, 3-30
- Desktop Controls, 3-4, 3-8, 3-29, 3-31
- devalias, 1-19, 1-22, 1-23, 2-10, 6-2**
- devfsadm, 13-12**
- device aliases, 1-17, **1-19**, 1-21, **1-22, 1-23, 16-10**
- device driver, 1-11, 6-1, 8-1, 8-6, 13-4, **25-8**
- device file, 8-6
- device files, 8-1, 8-6
- Device Management, 22-2**
- Device Security, 22-2**
- device tree, 1-13, **1-22, 1-23, 1-25, 2-36, 6-2**
- df, 26-5, 26-26**
- df command, 5-7, 15-9, 15-10*
- df -k command, 3-31, 7-34, 26-26*
- df -k command, 7-33*
- DFS, 14-10
- DFS architecture, 14-10
- dfshares Command, **14-1, 14-14**
- DHCP, xvi, 2-3, 2-12, 6-7, 6-8, **22-3**, 31-1, 31-2, 31-3, 31-5, 31-13, 31-16, 31-18, 31-25, 31-28, **31-29, 31-31, 31-33, 31-34, 31-35, 31-36, 31-37, 31-38**
  - acknowledgement packet, 31-3
  - agent, 31-5
  - assigned IP address, 31-3
  - BOOTP relay agent, 31-3
  - clearing a server, 31-36**
  - client, 31-14
  - client interface, 31-38
  - Command Line Tools, 31-32
  - command line utilities, 31-37
  - Configuration Wizard, 31-11
  - configure, 31-2
  - configure macros, 31-33
  - configure options, 31-33
  - data file, 31-5
  - DHCP manager, 31-28
  - dhcpinfo, 31-37
  - dhcptab table, 31-37
  - diskless clients, 31-1
  - DNS server, 31-14
  - domain, 31-14
  - Exporting DHCP Configuration Information, 31-28**
  - Exporting information, 31-28, 31-31 files, 31-4**
  - Importing Configuration Information, 31-28**
  - installation wizards, 31-5
  - K21dhcp, 31-4
  - K42inetsvc, 31-5
  - K43inetsvc, 31-4
  - K90dhcpagent, 31-5
  - macro, 31-24, 31-38
  - Macro, 31-21, 31-25
  - macros, 31-5
  - network address, 31-15
  - Network Address, 31-14, 31-15
  - network table, 31-5
  - NIS, 31-5
  - NIS+, 31-5
  - Number of IP Addresses, 31-19
  - Official Macro Classes, 31-24
  - options, 31-5
  - pool, 31-24
  - pool of IP address, 31-21
  - pool of IP addresses, 31-4
  - relay agent, 31-5, 31-6
  - relay agent setup, 31-6
  - renewal of the lease, 31-3
  - run control files, 31-4
  - S72inetsvc, 31-5
  - specify a router, 31-16
  - specify IP Addresses, 31-20
  - specify NIS, 31-16
  - specify NIS+, 31-16
  - Start the DHCP manager, 31-31
  - starting, 31-5
  - Starting from the command line, 31-36**
  - stopping, 31-5
  - Stopping from the command line, 31-36**
  - Subnet Mask, 31-14, 31-15
  - type of lease, 31-22
  - un-configure, 31-6
  - variables, 31-5
  - WWWservs, 31-26
- DHCP client, 31-2, 31-3, 31-23
- DHCP commands, 31-32
- DHCP Management, 22-2**
- DHCP manager, 31-2, 31-6
- DHCP Manager, 31-5, 31-6, 31-10
- DHCP packet, 31-3
- DHCP relay agent, 31-2, 31-6
- DHCP server, 31-1, 31-2, 31-3, 31-4, 31-5, 31-14, 31-19, **31-36**
  - DHCP values, 31-39
  - dhcpagent, 31-38
  - dhcpconfig, 31-2
  - dhcpconfig command, 31-32, 31-33
  - dhcpinfo command, 31-39
  - dhcptab table, 31-37
  - dhtadm command, 31-33, 31-37
  - dhtadn command, 31-33
  - diag-device, 1-21
  - diag-file, 1-21
  - diag-level, 6-3**
  - diag-switch?, 1-21, **6-3**
  - Directory Access Protocol, 30-3
  - Directory Console
    - banner, 30-16
    - status bar, 30-16
  - Directory Console, 30-15
  - banner, 30-15**



**fdfs**, 14-8, 15-2  
**ff** command, 15-9, 15-10  
fi, 12-18  
**Fiber optic cable**, 16-2  
File Descriptor File System. See **fdfs**  
File Manager, 2-38, 3-2, 3-3, 3-16, 3-17, 3-18, 3-19, 3-20, 3-24, 3-32, 3-34, 9-11, 9-12, 21-7, 21-8  
File Properties Utility, 3-1, 3-24, 3-25  
**File System Management**, 22-2  
**File System Security**, 22-2  
file system status, 15-6  
file system's status, 15-4  
`files.tar`, 29-16  
**filesystems**, 21-13  
**find**, 7-32  
*find* command, 7-32  
Find File Utility, 3-1, 3-22, 3-23  
Find Files Utility, 3-23  
**finger**, 16-7  
flashed, 1-12  
**flash-install**, 21-14  
floppy drive, 3-19, 3-20, 3-21, 3-22, 3-31, 3-32, 12-11, 14-2, 14-5, 14-9, 18-2, 18-8, 18-9, 21-6, 21-20  
**Font Administrator**, 3-32  
**for** command, 12-25, 12-26, 12-27  
**format**, 6-2, 22-1  
    **modify**, 26-19  
    partition, 26-19  
    **print**, 26-19  
Format, 21-8  
Format a Floppy Disk, 3-20  
**format** command, xv, 13-1, 13-7, 13-8, 13-12, 13-15, 14-1, 14-3, 15-2, 18-9, 21-3, 22-4, 26-18, 26-19  
**Format Floppy**, 3-20, 3-32  
Format Floppy window, 3-20  
Format window, 3-20, 3-21, 3-32  
Forth programming language, 1-12  
*FQDN*, 30-9  
fragment blocks, 15-4, 15-5, 15-6, 15-7, 15-8  
free data blocks, 15-3  
free file fragments, 15-3  
free inodes, 15-3, 15-6  
Front Panel, 3-1, 3-2, 3-3, 3-4, 3-5, 3-6, 3-7, 3-8, 3-9, 3-14, 3-15, 3-16, 3-18, 3-24, 3-27, 3-34, 7-31  
**fsck**, 26-5  
*fsck* command, 26-26  
fsck Command, 15-1, 15-11, 15-12  
**fsck Interactive mode**, 15-11  
fsck preen mode, 15-11  
fsck silent mode, 15-11  
**fstyp**, 14-10  
**ftp**, 27-6  
*FTP*, 2-28, 3-17, 7-19, 8-4, 10-1, 16-3, 16-5, 16-7, 16-14, 16-15, 18-12, 20-1, 20-2, 20-6, 21-29, 21-32, 27-2, 27-3, 29-10, 29-12  
**ftp** daemon. See **FTP**  
**FTP Management**, 22-2  
FTP session, 16-14

`ftp://ftp.rs.internic.net`, 29-8  
`ftp://ftp.rs.internic.net/domain/named.root`, 29-16  
*Fully Qualified Domain Name*. See **FQDN**

## G

**Gateway**, 16-7  
*gcore* command, 25-7  
**getprojidv**, 19-16  
**getprojnet**, 19-16  
GID, 7-2, 7-3, 7-7, 7-11, 7-15, 15-7, 22-8, 23-10, 25-5, 28-1, 30-33  
**GMT**, 2-4, 2-18, 21-28  
GNOME, 1-7, 3-1  
GNOME 2.0, 3-1, 20-4  
Graceful Shutdown, 2-1, 2-39, 3-1, 3-11  
Graphical User Interface. See **GUI**, See **GUI**, See **GUI**, See **GUI**  
*Greenwich Mean Time*. See **GMT**  
**grep** command, 5-7, 12-14, 14-2, 21-16, 21-24, 26-26  
group ID, 30-6  
Group ID. See **GID**, See **GID**, See **GID**  
**groupadd**, 7-5, 7-15, 30-6  
**groupdel**, 7-1, 7-16  
GroupID, 28-1, See **GID**  
**Groupname**, 7-3, 7-7  
**Group-password**, 7-3, 7-7  
**GUI**, 2-2  
**gunzip**, 3-17  
**gz**, 23-11  
**gzip**, 18-12, 18-15

## H

hacker, 1-17, 7-33, 10-2, 10-10, 14-6, 20-1, 20-2, 22-3, 23-6, 23-13, 24-5, 30-22, 30-59, 30-60  
Hackers, 7-20, 10-10  
**halt**, 25-10  
**halt** command, 3-12, 3-13  
hard drive components, 13-3  
**Hard Limit**, 7-30  
**Hard links**, 8-6, 8-7  
**Hardware Layer**, 16-9  
HCL, ii, 1-1, 3-2  
**head** command, 4-16  
Help Manager, 3-1, 3-4, 3-14, 3-15  
help with Solaris 9, 5-1  
hexadecimal  
    convert, 27-8  
hexadecimal digits, 27-8  
hexidecimal  
    hex numbers, 27-8  
home directory, 2-2, 2-25, 2-31, 2-38, 3-2, 3-3, 3-35, 3-36, 4-3, 4-4, 4-9, 7-1, 7-3, 7-6, 7-7, 7-8, 7-9, 7-12, 7-13, 7-18, 7-20, 7-21, 7-22, 7-23, 7-28, 7-30, 7-32, 7-34, 8-4, 8-7, 8-9, 12-27, 21-1, 22-4, 22-5, 22-9  
Home Folder, 3-4, 3-34  
host ID, 2-14, 16-8

**Host name, 19-10**  
 HostID, 1-17, 1-20, 16-13  
 HOSTID, 1-13  
 hostname, 14-11, **16-8, 21-6**, 21-18  
 Hostname, 2-3, 2-36, 12-2, 23-3, 29-11, 29-13, 30-5, 31-34, 31-35  
 hostname command, 29-14  
*hot spare*, 26-3  
*hot spare pool*, 26-4  
*Hot Spare Pool*, 26-24  
 hot swappable, 13-12  
 hot swappable disks, 26-1  
**hsfs, 14-9**, 15-1  
 HTML documentation, 3-16, 30-35  
 HTTP, 27-2  
**http://docs.sun.com, 5-1**  
**http://soldc.sun.com/support/drivers/hcl/index.html, 1-1**  
 http://store.sun.com, 1-1, 1-2, 1-4  
 http://sunsolve.sun.com, 3-35, **5-1**  
 http://sunstore.sun.com, ii  
**http://www.sun.com/bigadmin, 5-1**  
 http://www.sun.com/downloads, 30-4  
**http://www.teachmesun.com, ii**  
 httpd.conf, 29-19  
**HUB, 16-6**  
 HyperTerminal, 1-15, 1-16, **14-9**

## I

Icon Editor, 3-33  
 IDE controller, 13-4  
 IDE controllers, 13-6  
 idsktune utility, 30-6  
**if** control statement, 12-18  
 if/else if Tree, **12-20, 12-21**  
 ifconfig Command, **16-1, 16-10, 16-11**  
 in.dhcpd, 31-38  
 in.named, 29-20  
 in.named process, 29-16  
 in.named processes, 29-17  
**in.npd, 27-5**  
 incremental backup, 18-2, 18-4  
 index file, 23-4  
**inetd, 16-14, 16-15**, 21-29, 21-32  
 init, **1-18**, 2-6, **2-39, 2-40**, 3-2, 3-9, 3-10, 3-11, 3-12, 3-13, **3-22**, 3-35, 6-1, **6-2**, 6-4, 6-5, 6-6, 6-7, **6-8, 12-34**, 13-7, 14-5, **14-9**, 14-12, **14-13, 16-12**, 16-13, **18-1, 18-2**, 18-8, *18-18*, **19-11**, 21-20, 21-26, 21-29, **23-12, 23-19, 23-22, 23-23, 24-4, 24-5**, 24-6, **25-5, 26-26, 27-6, 28-3, 28-5, 28-8**, 28-18, 29-16, 30-54, 30-55, 30-56, 30-57, 30-58, 30-59, 30-60, 31-34, 31-35, **31-36**  
**init 0, 1-18**, 2-6, **2-39, 2-40**, 3-9, 3-10, 3-11, **6-2**, 16-13, 18-17, *18-18*, 21-20, **28-3**  
 init command, 3-10  
**initial\_install, 21-14**  
**Initialization, 7-5, 17-2**  
 initialization file, 7-5, 7-6

inode, 7-30, 7-32, **7-34, 7-35, 7-36, 8-6**, 8-9, **15-3, 15-4**, 15-6, 15-7, 15-10, 15-11  
*inode number*, 8-9, **15-3**, 15-6  
 inode table, 7-30, 8-9, 15-6  
**Inode Table, 15-3**, 15-4, 15-6  
 Insert/Append Mode, 4-5  
 install program, 21-8  
 Install server, 21-6  
**Install Server, 21-5**, 21-22  
**install\_cluster, 10-5**  
**install\_type, 21-11, 21-14**  
 Intel, ii, 1-1, 1-4, 1-6, **1-10**, 1-11, 1-16, **1-18**, 2-2, 3-2, 3-11, 3-17, 13-6, 20-3, **21-14**, 23-2, **23-9**, 23-10, 23-18, 29-1  
 Intel version of Solaris 9, 1-1  
**interfaces, 17-4**  
 International Standards Organization, 30-3  
**Internet Layer, 16-9**  
 Internet Service Provider, 27-1  
 inverse-domain-info, 29-9  
 IP, 27-2  
*IP address*, xvi, **2-2**, 2-4, 2-14, 2-15, 2-17, 2-19, 3-27, **14-13**, 16-4, 16-5, **16-7, 16-8, 16-10, 16-11, 16-12**, 16-13, **16-16, 17-2**, 20-7, **21-4, 21-5, 21-6, 21-11**, 21-18, 21-27, 21-29, 21-30, 21-31, 21-32, 23-3, **23-9, 23-10**, 23-13, 23-16, **23-18, 23-19, 23-20, 24-3**, 27-1, 27-2, 27-3, 29-1, 29-2, 29-3, 29-5, 29-6, 29-9, 29-10, 29-11, 29-12, 29-13, 29-15, 29-16, 29-17, 29-18, 29-19, 29-20, **30-13**, 30-27, 30-30, 30-39, 30-58, 31-1, 31-2, 31-3, 31-4, 31-18, 31-20, 31-24, 31-34, 31-35, 31-37  
 IP Address, 2-4, 2-13, 27-13, 29-5, 29-13, 30-5, 31-24, 31-25  
 IP packets, 27-2  
 IP part, 16-2  
*IP version 6*. See IPv6, See IPv6  
*iPlanet Console*, 30-15  
 iPlanet Directory, 30-2  
 IPv4, 27-1  
 IPv4/v6 routers, switches, 27-1  
 IPv6, 2-4, 3-26, 3-27, 16-1, 20-3, **23-6, 23-7, 27-1, 27-2**, 27-3, 27-4, 27-5, **27-6, 27-7, 27-8, 27-12, 27-13, 27-14**  
 Dual-stack” interface, 27-3  
 link-local address, 27-3  
 link-local prefix, 27-3  
 multicast group, 27-4  
 quality grade, 27-1  
 RFC 1886, 27-8  
 RFC 1933, 27-8  
 router solicitation, 27-4  
 stateful address, 27-4  
 stateless address, 27-4  
 stateless address auto configuration, 27-4  
 IPv6 address, 27-2  
**IPv6 Addressing, 27-3**  
 IPv6 header, 27-2  
 ISO, 30-3  
 ISP, 27-1

## J

Java Runtime Environment. *See* JRE  
JRE, 30-3  
jumper, 13-3  
Jumpstart Boot server, 21-6  
JumpStart client, 21-5, 21-6  
**JumpStart Client, 21-4**  
JumpStart configuration files, 21-22  
JumpStart Configuration server, 21-6  
JumpStart Install server, 21-6  
JumpStart installation, 21-2, **21-6**  
JumpStart Installationserver, 21-26  
JumpStart server, 21-23  
**JumpStart Servers, 21-4**  
`jumpstart_sample` directory, 21-10

## K

K scripts, 12-3  
K22acct, 28-3  
Kerberos, 2-4, 2-15, 2-16  
*Kerberos security server, 2-15*  
Kernel, 6-1, **6-3**, 6-4, 6-6, **7-5**, 8-1, 8-2, 8-4, 12-2, 13-10, 13-11, 21-8, 21-25, 24-1, **24-3**, 26-13  
Kernel module, 6-3  
**key, 23-3**  
*kiosk, 2-21, 14-5*  
*kiosk display, 2-21*  
Korn shell, 3-36, **7-23, 7-25, 12-2**  
Korn Shell, 12-1, 12-2  
*ksh* command, 12-2

## L

**LAN, 16-1, 31-15**  
languages, 1-6  
**largefiles, 14-6**  
**Last Boot Time, 19-10**  
**lastdate, 28-6**  
LDAP, xvi, 2-4, 2-17, 7-29, 7-30, 16-8, 17-6, **19-17**, 20-2, 20-3, 20-6, 20-7, 20-10, 21-28, 21-29, **22-3**, 28-12, 29-2, 29-3, 29-4, 29-5, 30-1, 30-2, 30-3, 30-4, 30-6, 30-10, 30-11, **30-12**, 30-13, 30-18, 30-22, 30-23, 30-24, 30-27, 30-28, 30-29, 30-32, 30-33, 30-34, 30-35, 30-36, 30-37, 30-38, 30-39, 30-40, 30-41, 30-42, 30-43, 30-46, 30-47, 30-49, 30-50, 30-51, 30-52, 30-54, 30-55, 30-56, 30-58, 30-59, 30-60, 30-61, 30-62, 30-63, 30-64  
Administration Domain, 30-16  
Administration Server, 30-2  
attributes, 30-3  
central administration, 30-2  
Console, 30-2  
documentation, 30-4  
**Features, 30-2**  
hierarchical namespace, 30-2  
hierarchy, 30-1  
history, 30-3  
LDAP clients, 30-2  
main components, 30-4

Master Servers, 30-2  
memory, 30-4  
multi-master, 30-2  
Multi-Master, 30-5  
object classes, 30-3  
objects, 30-3  
organizational unit, 30-1  
Replica Servers, 30-2  
replication, 30-2  
Requirements, 30-4  
v3 protocol, 30-1  
X.500 directory, 30-3  
LDAP Console, 30-3  
LDAP Master Server, 30-5  
LDAP server, 30-1, 30-2  
LDAP version 2, 30-3  
LDAP version 3, 30-3  
LDIF, xvi, **30-12**, 30-34, 30-37, 30-41, 30-42, 30-43, 30-50, 30-51, 30-52, 30-63  
lease time, 31-3  
Lempel-Ziv compression, 18-15  
license, 1-8  
**License Fees, 1-9**  
Lightweight Directory Access Protocol. *See* LDAP  
**lineuse, 28-7**  
Linux, 23-9, 23-10  
Live Upgrade installation, 21-2  
**Live Upgrade Method, 2-3**  
**ln, 8-8**  
`ln` command, 28-3  
*`ln -s` command, 8-9*  
**Local Area Network.** *See* LAN  
LOCAL authentication, 23-14  
local disk set, 26-3  
local file systems, 14-1, **14-8, 15-1**  
Local JumpStart, 21-7  
Local JumpStart installation, 21-5  
**Local printer, 17-2**  
Local-Area Network, 31-15, *See* LAN  
**locale, 21-14**  
Lock icon, 3-5, 3-8  
lock symbol, 3-3  
**log, 28-6**  
**Log Management, 22-3**  
Log Viewer, 19-12  
Log Viewer 4.1, 19-9, **19-12**  
`logger` command, 24-1, 24-6, 24-7  
**logging, 7-31, 14-6, 15-4, 19-13, 20-5, 20-8, 24-5, 25-5, 26-7, 30-23, 30-37**  
**logging device, 26-8**  
Logging TCP Connections, 24-5  
*logical device name, 13-10, 14-3, 14-4, 14-6, 15-9, 15-10, 26-13*  
*logical volume, 26-1*  
Login Manager, 2-36, 2-37, **2-40**, 3-1, 3-2, **3-3**, 3-7, 3-9, 3-11, 3-35, 3-36, 6-8  
**loginID, 7-2, 7-3, 7-4, 7-6, 7-7, 7-8, 22-5, 22-6**  
**LoginID, 7-4, 7-6, 7-8**  
**loginlog, 28-6**

**Login-shell**, 7-3, 7-7  
 loopback-domain-info, 29-9  
**lost+found** directory, 21-8  
**lp**, 17-3  
*lp* command, 24-4  
**lpadmin**, 17-3, 17-11  
**LPDEST**, 17-12  
**lpfilter**, 17-3  
**lpforms**, 17-3  
**lpget**, 17-3  
**lpmove**, 17-3, 17-11  
**lpr**, 17-11  
**lpsched**, 17-1, 17-3, 17-4, 17-9  
**lpsched** process, 17-2, 17-4, 17-9  
**lpset**, 17-3  
**lpshut**, 17-3  
**lpstat**, 17-3, 17-11  
**lpsystem**, 17-3  
**lpusers**, 17-3  
**ls**, 4-2  
**ls** command, 4-3, 4-4, 5-4, 5-6, 8-3, 8-9, 8-10, 12-13, 12-14, 12-35, 14-13, 18-10, 30-51  
**ls -l** command, 7-10

## M

M4, 24-4  
**M4** Macro processor, 24-4  
 M4 Macro Processor, **24-4**  
 M4 processor  
   **ifdef** keyword, 24-4  
 M4 processor, 24-4  
**MAC Address**, 16-6  
 magnetic head, 13-5  
**Mail Management**, 22-3  
 Main Panel, 3-3  
**Maintenance and Repair**, 22-3  
 major number, 8-6  
*make* command, 23-11  
 make file, 23-5  
**Makefile**, 23-7, 23-11, 23-13  
**man** pages, 2-2, 5-1, 5-2, 5-4, 5-5, 5-6, 5-7, 5-8, **8-5**, 21-21  
 Man pages, 5-1, 5-4  
**MANPATH**, 5-4, 5-5, 5-7, 12-8  
**MANSECTS**, 5-4  
**map**, 23-3  
 map files, 23-8  
**map.key.dir**, 23-4  
**map.key.pag**, 23-3  
 master disk, 13-6  
 Maximize button, 3-8  
 media kit, 1-4, 1-5, 1-9  
 media kit release date, 1-4  
 media kits, 1-4, 1-5, 1-7, 1-8, 2-21, 9-11  
**metadb**, 26-8  
**metadb** command, 26-8, 26-12, 26-13, 26-14  
*metadevice*, 26-2

*metadevice* name, 26-23  
*metadevice* names, 26-2  
*metainit* command, 26-25  
 minor number, 8-6  
**mkdir**, 3-17  
**mkdir** command, 4-2, 13-11, 14-4  
**mkfile**, 25-3, 26-26  
*modem*, 1-10, 1-13, 1-15, 1-16, 2-5, 2-7, 2-8, 2-37, 3-17, 3-31, 4-5, 8-1, **8-6**, 16-6, 18-12, 19-4, 31-15  
 modem bank, 28-2  
**monacct**, 28-7, 28-15  
 monitor program, 1-10, 6-1  
 Monthly Command Summary, 28-15  
**more** command, 4-16, 4-17, 6-9, 17-8, 20-9  
**mount** command, 4-2, 13-8, 13-11, **14-1**, 14-2, 14-3, 14-6, 14-7, 14-9, 14-10, 14-12, 14-13, **14-14**, **14-15**, 26-19, 26-26  
*mount point*, 4-2, **8-4**, 14-1, 14-2, 14-4, 14-6, 14-7, 14-9, 14-10, 15-4, 15-9, **15-11**, 18-6, 18-7, 18-8, 21-13, 26-19, 26-26, 26-27  
 mouse Double-Click speed, 3-30  
 Move handles, 3-5  
**Multicast Address**, 27-3  
*mv* command, 6-9

## N

*Name Servers*, 16-8  
*name* service, 2-16, 2-17, 16-8, 21-28, 21-30, **22-3**, 29-5, 31-34  
 Name Service, xvi, 2-4, 2-16, 2-17, 7-29, **16-7**, 16-8  
**Name Service Management**, 22-3  
 named. root, 29-8  
 named. root, 29-16  
 NAT, 27-1  
**NEC DOS**, 3-20, 3-32  
*Netmask*, 2-14  
 Netscape icon, 3-27  
 Netscape Navigator, 3-2, 3-4, 3-5, 3-27, **7-30**, 7-31, 7-32, 20-6, 29-2  
**netstat** command, 27-7  
 network address, 2-14, 16-1  
 Network Address Translation. *See* NAT  
 network broadcast address, 31-1  
**Network Cables**, 16-1  
 Network File System. *See* NFS, *See* NFS  
 network ID, 16-8  
 Network Information Server, 23-1, *See* NIS  
 Network Information Services. *See* NIS  
**Network Interface Card**. *See* NIC, *See* NIC  
**Network Interface Layer**, 16-9  
 Network layer, 16-4, 16-7, 20-1, 20-2  
**Network Layers**, 16-2  
**Network Media**, 16-1  
 Network Mounting Options, **14-14**  
 network name, 14-11  
**Network printer**, 17-2  
 Network Protocols, 16-2

network topology, 31-1  
 Networked File System. See NFS, See NFS  
*networked printer*, 17-7  
*networked printers*, 17-6

**newfs**, 3-21, 3-22, 14-3, 14-4, **14-15**, **15-1**, 15-2, **15-12**, **15-13**, **18-1**, 18-7, 18-9, 18-16, **21-8**, **25-3**, 26-3, **26-20**, **26-21**, 26-26, 26-28, 26-30, **26-34**

**newfs** command, 14-3, **15-12**, **21-8**, **26-18**

**nfs**, 14-9

**NFS**, 6-4, 7-7, **8-3**, 8-4, 14-10, **14-11**, 14-12, **14-13**, **14-15**, 15-1, **15-13**, **21-11**

**NFS client**, 14-10, **14-15**

NFS file systems, 6-4, 15-1, **15-13**

**NFS server**, 7-7, **8-3**, 14-10, 14-12, **14-13**

**NIC**, **16-6**, **16-10**, **16-11**, **23-4**

NIC card, 16-6

NIS, xvi, 2-4, 2-17, 3-23, 7-29, 7-30, **14-11**, 16-8, 17-6, **17-12**, **19-17**, 20-7, **21-5**, 21-27, 21-28, 21-29, 21-30, **22-3**, **23-1**, 23-2, 23-3, **23-4**, **23-5**, **23-6**, **23-8**, **23-9**, **23-10**, 23-11, **23-12**, **23-13**, **23-14**, 23-16, **23-17**, **23-18**, **23-19**, **23-20**, **23-21**, **23-22**, **23-23**, 28-12, 29-2, 29-3, 29-4, 29-5, 30-2, 30-33, 30-61, 31-33

NIS, 23-15

NIS client, 23-3, **23-10**, 23-13

**NIS Client**, **23-3**, 23-11

NIS clients, 23-2, 23-5, 23-22

NIS Clients, 23-6

NIS domain, 23-2, **23-3**, 23-8

NIS files, 23-2

NIS map, 23-2, **23-5**

NIS map file, 23-2

NIS Map Files, 23-3

NIS maps, 23-3, 23-13

NIS Master, 23-6

NIS Master server, 23-1, **23-6**, **23-8**, 23-10

**NIS Master Server**, **23-2**

*NIS namespace*, 23-2

NIS primary key, 23-3

NIS server, 23-2, **23-5**, **23-6**, **23-8**, **23-9**, **23-10**, 23-18

**NIS Server**, **23-12**

NIS services, 23-10

NIS Slave, 23-3

NIS Slave server, 23-2

**NIS Slave Server**, **23-2**, 23-3

NIS slave servers, 23-1

NIS Slave servers, 23-2

NIS slaves, 23-5

NIS Slaves, 23-6

NIS tables, 23-10

NIS+, 7-29, 7-30, **14-11**, 16-8, 17-6, 20-7, **21-5**, 21-27, 21-28, 21-29, 21-30, **22-3**, **23-1**, 23-2, 23-5, **23-13**, 23-14, 23-16, **23-17**, **23-18**, **23-20**, **23-21**, **23-22**, **23-23**, 28-12, 29-2, 29-4, 30-2, 30-33, 30-61, 31-33

Commands, **23-17**

**Nisaddcred**, 23-17

**nisaddent**, 23-17

**nisauthconf**, 23-17

**nisclient**, 23-17

**nisinit**, 23-17

nismaster, 23-17

**nissserver**, 23-17

Replica server, 23-17

Replica servers, 23-16

Setup, 23-18

NIS+ client, 23-18

NIS+ clients, 23-13

NIS+ directory object, 23-15

NIS+ Master server, 23-16

*NIS+ name space*, 23-13, **23-15**

NIS+ Name Space, **23-14**

NIS+ Replica server, 23-17

NIS+ replica servers, 23-13

NIS+ Root Master server, 23-15

NIS+ server, 23-18, **23-22**

NIS+ table, 23-13, 23-15

NIS+ tables, 23-14

NIS+ Tables, **23-14**

NIS+master, 23-13

NIS+server, 23-18

**nisbackup Command**, 23-21

niscat, **23-20**

**niscat** command, 23-22

**nisdefault command**, 23-21

**nismaster**, 23-17

**nispopulate**, 23-18

*nispopulate command*, 23-19

**nisrestore Command**, 23-21

**nissserver**, 23-18

**nisstat Command**, 23-21

**nolargefiles**, 14-6

**nologging**, 14-6

**nosuid**, 14-6

**nsswitch.conf**, 23-8

**nsswitch.conf** file, 29-4

null modem cable, 1-10, 1-13, 1-15, 1-16, 2-7

nulladm, 28-18

**nvalias**, 1-22, **1-23**

nvedit, **1-23**

NVRAM, 1-11, 1-16, 1-17, **1-19**, 1-20, **1-21**, 1-22, 16-13

NVRAM chip, 1-17

Nvramrc, 1-21

## O

Obdiag Utility, **1-24**

**Object Access Management**, 22-3

oem-banner, 1-21

oem-banner?, 1-21

oem-logo, 1-21

OK, 1-10, 1-12, 1-16, 1-17, **1-18**, 1-20, 1-21, 1-22, 1-23, **1-24**, 1-25, 1-26, 2-1, **2-2**, 2-5, 2-6, 2-7, 2-8, 2-23, 2-24, 2-32, 2-36, 3-10, 3-11, 3-23, 3-24, **3-25**, 3-29, 3-31, 3-34, 6-2, **6-3**, 7-17, **7-37**, **9-9**, **10-3**, 13-4, 13-7, **16-12**, 16-13, **16-15**, **17-10**, **17-11**, 18-17, 21-1, **21-4**, **21-5**, 21-6, 21-8, 21-20, 21-21, 21-27, 21-31, 21-32, **22-11**, **22-16**, **25-10**, 26-27, **30-14**, 30-21, 30-23, 30-24, 30-26, 30-31, 30-41, 30-42, 30-46, 30-48, 30-49, 30-53, 30-54, 31-18, 31-23

OK prompt, 18-18

OK prompt, 1-10, 1-12, 2-5, 2-6, 2-7, 3-11, 21-2, **21-4**, 21-6

on-board disk controller, 13-4, 13-5

open a webpage, 3-27

Open Floppy, 3-19, 3-20, **3-32**, **21-8**

Open Log File, 19-13

Open Systems Interconnection. *See* OSI

OpenBoot, 1-10, 1-11, 1-12, 1-13, 1-15, 1-16, **1-18**, 1-20, **1-22**, **1-23**, 1-24, 2-5, 2-6, 2-10, 2-36, 3-11, **6-1**, 6-2, **6-3**, 13-4, **16-12**, 21-2, **21-12**, 21-21

OpenBoot help, 1-19

OpenWindows, 3-1, **8-5**, 31-32

Oracle, 19-15

OSI model, 16-3, **16-4**, **16-5**, **16-6**, **16-7**, **16-9**, 20-1, 27-2

## P

**Packet**, 16-2

**pag**, 23-3

partition, xv, **1-18**, 2-1, 2-10, 2-25, 2-31, 2-32, 3-13, 13-1, 13-2, **13-8**, **13-11**, **13-13**, **13-14**, **13-15**, 14-1, 14-6, **14-7**, **14-8**, 15-2, 15-3, **15-4**, 15-5, 15-6, 15-9, 20-5, **21-15**, 21-16, **21-17**, **25-1**, **25-8**, 26-1, 26-4, **26-5**, 26-6, 26-8, **26-17**, 26-19, **26-20**, 26-24, **26-31**

**passwd**, xv, **2-37**, 3-23, 3-24, **4-17**, **5-6**, 7-2, **7-3**, **7-4**, **7-6**, 7-8, 7-11, **7-25**, 7-27, 7-29, 7-30, **7-31**, **7-34**, 8-3, 8-10, 12-2, 15-6, 15-7, **17-9**, 17-12, 22-6, **22-16**, **23-5**, **23-9**, 23-14, 23-22, 28-12, 29-3, 29-4, **30-11**, 30-33

patch, 10-1

patch back-out directory, 10-5

**patch cluster**, 5-3, 10-1, 10-4, 10-5

Patch Manager, **10-6**

patch number and a revision number, 10-1

Patch Structure, **10-6**

**patch\_order**, 10-5

**patchadd**, 10-2, 10-5

**patchrm**, 10-2, 10-5, 10-6

**pcfs**, 14-9, 15-1

**PCFS (DOS)**, 3-20

PDF documentation, 3-16

**pdf** files, 3-16, **5-1**, 5-2

PDF format, 30-4

peer to peer networks, 16-6

Performance Manager, 19-14

Performance Manager 1.0, **19-13**

**pfcs**, 22-5

**pfinstall**, 21-17

**pfinstall** command, 21-7, **21-17**

**pfinstall** program, 21-18

**pfinstall** Utility, 21-17

**pfksh**, 22-1, 22-5

**pgrep**, 25-7, 29-18

*physical device name*, 13-10, 21-12

**Physical Layer**, 16-5

PID, 6-4, 11-7, **23-12**, **25-7**, 29-17, 30-29, 30-30

**ping**, 16-7, **23-9**, **29-1**, 29-6, 29-13

**ping** command, 16-13, **23-10**, 24-6, **27-6**, **27-7**, 30-5, 31-2

pipes, 12-2, 12-14

**pkgadd**, 3-16, **9-3**, **9-4**, **9-6**, 9-7, **9-9**, **9-10**, **22-11**, **23-19**, 28-3

*/cdrom/cdrom0/Solaris\_9\_sparc/s0/Solaris\_9/Product*, 23-19

**pkgask**, 9-3

**pkgchk**, 9-3, **9-4**, **9-5**, **9-6**

**pkginfo**, 9-2, **9-3**, **9-6**, **10-6**, 28-3, 28-18

**pkgmap**, 9-3, **10-6**

**pkgparam**, 9-3, **9-5**, **10-2**

**pkgrm**, 9-3, **9-4**, **9-6**

**kill**, 3-35, 16-15, **23-12**, 29-17, **29-20**, 30-30

**pkunzip**, 10-4

*platter*, 13-4, 13-5, 13-6, 15-2

**pntadm** command, 31-33, 31-34, 31-35

port 111, 16-15

POSIX style print command, 17-12

POST, 1-13, 1-16, 2-5

post-installation shell scripts, 21-2

Power Management, 2-40, 3-31, **3-32**

Power On Self Test. *See* POST, *See* POST, *See* POST

**power-off**, 1-16, 1-20, 1-22, 3-10, 3-11, 13-7

**poweroff** command, 3-13

Preinstall Boot Image, 21-3

pre-installation shell scripts, 21-2

**Presentation Layer**, 16-3

**PressKey** variable, 12-36

*primary group*, 7-2, **7-7**, 7-20, **7-22**, 7-27

primary IDE interface, 13-6

primary NIS+ tables, 23-14

**Print Administrator**, 3-32

**Print client**, 17-2

Print Manager, 17-1, 17-2, **17-5**, 17-6, **17-10**, 22-2

**Print server**, 17-2

**printenv**, 1-16, **1-18**, 1-19, 2-6, 6-2

**PRINTER**, 17-12

**Printer Management**, 22-3

Printer names, 17-2

printer ports, 17-7

**probe-fcal**, 1-20

**probe-ide**, 1-20, 13-4

**probe-scsi**, 1-20, 13-4

**probe-scsi-all**, 1-20

Process File System. *See* procfs

Process IDentification Number. *See* PID

**Process Management, 22-3**  
 Process manager, 11-6  
 Process Manager, 3-2, 3-7, 11-1, 11-6, 11-8, 19-9, **19-10, 19-11**  
**proafs, 13-1, 14-8, 15-2**  
**procfs** file system, 13-1  
 prof\_attr, 22-7  
 Profile Bourne Shell, 22-4, *See* psh  
 Profile C Shell, 22-5  
**profile diskette, 21-5, 21-7**  
**Profile file, 21-5**  
 Profile Korn shell. *See* pksh  
 Profile Korn Shell, 22-5  
 profile text file, 21-7, 21-11  
**profiles, 22-5, 22-6**  
 Profiles, 22-4  
**profile-text-file, 21-7**  
 Project, 19-14  
 Project Database Manager 1.0, **19-16**  
**Project Management, 22-3**  
**project.byname, 19-17**  
**project.bynumber, 19-17**  
**ProjectName, 19-16**  
 Projects view, 19-14  
**PROJID, 19-16**  
 PROM Chip, **1-16**  
 PROM firmware, 1-16  
 Proxy Server, 2-4, 2-19  
 prstat command, 11-7, 11-8  
**prtconf, 13-7, 13-12**  
**prvtoc, 21-17, 21-18**  
**prvtoc** command, 15-5, **21-17**  
 ps, 29-17  
*ps* command, 12-2  
 pseudo file systems, 15-1  
**psh, 22-1, 22-4**  
**pwd, 7-12**  
 pwd command, **4-2, 4-3, 7-13, 8-8**

## Q

**Queuing, 17-2**  
**quot** command, 15-10  
 quota, 7-1, 7-30, **7-31, 7-32, 7-33, 7-34, 7-35, 7-36**  
**quotacheck, 7-31, 7-33**  
**quotaoff, 7-34, 7-35, 7-36**  
**quotaon, 7-33, 7-34, 7-35, 7-36**  
**quoton, 7-31**

## R

RAID 0, xvi, **19-4, 20-5, 26-1, 26-4, 26-5, 26-7, 26-17, 26-21, 26-23, 26-30, 26-31, 26-32**  
 ConcatenatedVolume, 26-21  
*create device, 26-22*  
 RAID 0 Concatenated Volumes, 26-4  
 RAID 0 Striped Volumes, 26-4

RAID 1, xvi, 20-5, **26-1, 26-4, 26-5, 26-31, 26-32, 26-33, 26-34, 26-35**  
 mirror, 26-4  
**RAID 1 Volume, 26-31**  
 RAID 5, xvi, **19-4, 20-5, 26-1, 26-4, 26-5, 26-7**  
 parity information, 26-5  
 RAID Levels, **26-4**  
*RAID Volumes, 26-2*  
 raw device file, 26-2  
 raw mode, 8-6, 13-2, 14-5, **18-2**  
 RBAC, 3-10, **7-16, 7-18, 20-8, 22-1, 22-2, 22-6, 22-11, 22-12, 22-16, 23-7**  
*RBAC database files, 22-6*  
**rcp, 16-7, 27-6**  
**rcs.d, 12-3**  
**read** command, 12-10, 12-11, *12-16*  
 read permission, 7-9  
 read/write permissions, 7-9  
**reboot** command, 3-13  
**reboots, 28-7**  
 recommended patch cluster, 10-1, 10-4  
*redirection., 12-12, 12-14*  
 reformat, 2-30, 3-20, 13-1, **25-1, 26-18**  
 reformat a disk, 3-20  
 Registration, 2-38  
 Regular files, 8-9  
 Remote Login, 3-2  
**Remote printer, 17-2**  
*Remote Procedure Call. See RPC*  
 Remote Procedure Calls. *See* RPC, *See* RPC  
 removable media, 3-19, **14-5**  
 Removable Media Manager, 3-20, 3-21, **3-32, 21-8**  
 Remove NIS, **23-11**  
 repair the CDE, 3-35  
**Repeater, 16-6**  
**repquota, 7-1, 7-31, 7-34, 7-35**  
**re-preinstall, 21-1**  
**re-preinstall** script, 21-2  
**reset-all, 1-19, 1-20, 6-2, 6-3**  
**resolv.conf, 5-4, 29-2, 29-6, 29-20**  
**Return code 2, 10-5**  
 reverse lookup, 29-9  
**rexec, 16-7**  
 rights, 22-2  
**Rights Delegation, 22-3**  
 RJ-45, 16-6  
**rlogin, 3-33, 16-7**  
**rm, 23-12**  
*rm* command, 7-6, **7-36, 9-4**  
 roaming users, 31-1  
 Role Based Access Control. *See* RBAC, *See* RBAC, *See* RBAC  
 RBAC, *See* RBAC, *See* RBAC, *See* RBAC  
**roleadd, 22-5, 22-16**  
**roledel, 22-5**  
**rolemod, 22-5**  
**roles, 22-5, 22-6**  
 root ( / ) file system, 7-33, **7-36, 13-1, 15-12, 18-6, 18-18**

root (/) directory, 3-18  
**root** account, 19-15  
 Root Master NIS+ server, 23-15  
 Root Master *Replica NIS+ servers*, 23-15  
 root Master Server, 23-13  
 Root Master server  
   NIS+, 23-16  
 root servers, 29-8  
 root slice, 26-8  
 root user, 22-1  
**ROOT USER, 2-37**  
**root\_device, 21-15**  
 router, 1-10, 2-4, 2-14, 2-15, 16-1, 16-3, 16-4, 16-5, **16-10**, **16-12**, **21-22**, 27-3, 29-5, 29-18, 30-5, 31-37  
   IP address, 31-15  
   router advertisements, 27-4  
**Router, 16-7**  
 router discovery protocol, 31-15  
*router IP address, 2-15*  
 routers  
   prefix information, 27-4  
*RPC, 2-8, 2-36, 14-10, 16-15, 23-14, 23-19, 29-12*  
*RPC daemon, 16-15*  
**rpirt<MMDD>, 28-6**  
**rsh, 16-7, 21-12, 27-6**  
**rules, 21-7, 21-22, 21-23, 21-26**  
**rules** file, 21-7, 21-8, 21-10, 21-11  
**Rules file, 21-5**  
**rules** files, 21-7  
**rules.ok, 21-5, 21-6, 21-7, 21-8, 21-18, 21-22, 21-23, 21-26**  
**rules.ok** file, 21-23  
 run control script, 6-4, 6-5, 6-8, 6-9, **12-34**, 30-54  
 run control scripts, 6-1, **6-4**, 6-5, 6-7, 6-9, **8-3, 9-4, 12-34**, 12-36, 28-3, **28-8**, 30-64  
 Run Control Scripts, 6-1, 6-5, 6-8, *12-3*, **12-34**, 30-1, 30-54  
 run level, 1-12, 3-10, **3-12**, 3-13, **3-35**, 6-1, **6-4**, 6-5, 6-6, 6-7, 16-12, 16-13, 26-12, **26-35, 26-36, 28-3, 28-7**  
 run levels, 3-11, **3-12, 6-4**, 6-5, 28-1, 28-3  
 runacct, 28-5, **28-7, 28-16**  
**rup, 16-7**  
**rwho, 16-7**

## S

S scripts, 12-3  
 S22acct, 28-3  
**s99dtlogin, 3-35, 6-8**  
**sag, 28-2**  
 Samtron, 2-6  
**sar, 28-2**  
**savecore** command, 25-9  
 sbus-probe-list, 1-22  
 Scalable Processor Architecture. See SPARC  
**sched** process, 6-4  
 screen saver, 3-30  
 SCSI, 13-6

SCSI controller, 2-29, 13-4, 13-9, 13-10  
 search for files, 3-22, 3-23  
 second level indirect pointers, 15-7  
 secondary IDE interface, 13-6  
*sector, 6-3, 13-6, 13-13, 15-2, 15-3, 15-4, 15-5*  
*sectors, 13-6, 13-13, 15-2, 15-3, 15-4, 15-5, 15-6, 15-13, 26-6, 26-13*  
 Secure Sockets Layer. See SSL  
 security-mode, 1-22  
 security-password, 1-22  
**sed, 4-17**  
 serial port A, 1-10, 1-21, 1-22, 2-7  
 Serial Port A, 1-10, 1-11, 1-12, 1-16, **1-18**, 3-9, 3-10  
**Session Layer, 16-4, 16-7**  
 Session Manager, 3-2, **3-3**, 3-35, 3-36  
**set +x, 12-32**  
**set -x, 12-32**  
**set-defaults, 1-19**  
**setenv, 1-16, 1-19, 1-20, 1-22, 1-23, 1-24, 1-25, 2-6, 6-2, 6-3, 21-21, 21-22**  
**setup\_install\_server, 21-5, 21-6, 21-24, 21-25**  
 Shadow inode, 15-7  
**share** command, 14-1, **14-10, 14-11, 14-12, 14-15, 21-31**  
 share Command, **14-10**  
 shareall, **14-13**, 21-26  
 shared directory, 14-10, **14-11**, 14-12, 15-1, **21-26**  
 sharing a directory, 14-10  
 Shell scripts, 12-1  
**showrev, 10-2**  
 shut down Solaris, 2-38, **2-39**, 3-10  
**shutdown, 2-38, 3-9, 3-10, 3-11, 3-12, 3-13, 6-1, 6-5, 21-20, 28-3, 28-4, 28-5, 28-10, 28-11, 30-29, 30-54, 30-55, 30-57, 30-58**  
**shutdown** command, 3-10, 3-11, **3-12**  
*single indirect pointer block, 15-7*  
 slave disk, 13-6  
*slice, 1-23, 2-10, 2-11, 2-31, 2-32, 2-36, 7-30, 7-32, 13-1, 13-2, 13-3, 13-6, 13-8, 13-9, 13-10, 13-11, 13-14, 13-15, 14-1, 14-2, 14-3, 14-4, 14-6, 14-7, 14-8, 15-2, 15-3, 15-4, 15-6, 15-10, 15-12, 18-7, 18-8, 18-9, 18-10, 18-16, 18-18, 21-2, 21-3, 21-11, 21-13, 21-14, 21-15, 21-16, 21-17, 21-18, 25-1, 25-2, 25-3, 26-4, 26-6, 26-8, 26-9, 26-10, 26-13, 26-14, 26-15, 26-17, 26-18, 26-19, 26-20, 26-21, 26-24, 26-28, 26-29, 26-30, 26-31, 26-34  
 slice 1, 25-1  
 Slice 1, 2-31, 13-2, See swap partition  
 slice 2, 13-2  
 slice 7, 21-4  
 Slim Kit, 1-2, 1-5, 1-6, 1-7, 1-8, 2-3, **5-1**  
 Small Computer Systems Interface. See SCSI  
**Smart Card, 3-32**  
 SMC, 2-3, **7-3, 7-18, 7-24, 17-6, 19-1, 19-3, 19-4, 19-5, 19-6, 19-7, 19-9, 19-13, 19-17, 20-3, 20-5, 20-8, 20-9, 22-1, 22-12, 22-17, 26-1, 26-2, 26-6, 26-7, 26-8,***



- su log, 22-4
- subnet mask, 21-6, 27-2, 31-1
- Subnet Mask, 2-4, 29-5, 30-5
- subpanel, 3-4, 3-6
- subpanel menu, 3-4
- subpanels, 3-3
- sudo, 22-1**
- suid, 14-6**
- Sun keyboard, 1-3, 1-11, 1-12, 6-2, **11-5**, *12-14*, *12-15*, 12-26, 21-31, 26-26
- Sun Microsystems, ii, xvi, 1-1, 1-2, 1-4, 1-5, 1-7, 1-8, 1-10, 1-11, 1-17, **1-25**, 2-8, 2-21, 2-29, 2-35, 2-37, 2-40, 3-1, **3-3**, 3-27, 3-31, 5-1, 6-1, 9-2, 10-5, 15-2, 20-3, 20-4, 20-6, 20-7
- Sun monitor, 1-3
- Sun mouse, 1-3
- Sun One Directory server, 1-8
- Sun One Directory Server, xvi
- Sun ONE Directory Server, 30-1, 30-2, 30-3
- Sun VAR, 1-4
- SunBlade 100, ii, 1-2, 1-4, **1-12**, **1-20**, **1-24**, **1-25**, 1-26, 2-4, 2-9, 7-29, 8-10, **16-11**, **21-17**, 26-14, **26-15**, 26-21, 29-18
- SunInstal, 21-10
- Suninstall, 2-3**, 2-7
- SunInstall, 21-6**
- SunInstall** program, 21-19
- Suninstall script, 2-3**
- SunScreen, 22-3**
- SunService contracts, 10-1
- sunsolve.sun.com, 3-35, 5-3, **5-8**, 10-1, 10-5, **28-19**, **29-2**, 29-19, 30-58
- Sunstation 2, 1-12
- Sunstation 5, 1-12
- SuperBlock, 15-3**, **15-4**, 15-5, 15-6
- Suspend System, 3-13, **3-32**
- SVGA, 1-4
- SVM, 26-1, 26-2, 26-26, **26-35**
- swap Command, **25-1**
- swap file, 25-1*
- Swap File System. See swapfs
- swap** partition, 2-10, 2-31, **13-1**, 13-2, **25-1**, 25-2
- swap slice, 26-8**
- swap space, 25-1
  - Managing, 25-1
- swapfs, 14-8**, 15-2
- Switch, 16-6**
- symbolic (soft) links, 8-7
- symbolic device files, 8-1
- symbolic link, 8-6
- symbolic links, 8-7, 28-3
- symbolic or soft link, 8-7
- symbolic slice, 13-1, *18-18*, See slice 2
- symbolic, or soft, link, 8-7
- sync command, 25-10**
- sys.dtpofile, 3-3**, 3-35, 3-36
- sysidcfg, 21-5**, **21-6**, **21-7**, 21-22, **21-26**
- sysidcfg** file, 21-18, 21-19

- syslog
  - level, 24-2**
- Syslog Error Messages, **24-1**
- syslog** file, 19-12
- syslog.conf
  - action field, 24-2
- syslog.conf
  - facility.level, 24-2
- syslog.conf
  - mail.crit, 24-2
- syslog.conf
  - user.err, 24-2
- syslogd, 19-13**
  - level, 24-3**
- syslogd** daemon, 24-1, **24-3**, **24-4**
- system accounting
  - acctcon, 28-5
  - CPU usage, 28-1
  - CPU Usage, 28-1
  - dodisk, 28-5
  - fwtmp, 28-5
  - monacct, 28-4
  - prctmp, 28-5
  - prtacct, 28-5
  - runacct, 28-5
  - shutacct, 28-5
  - turnacct, 28-5
  - wtmpfix, 28-5
- system accounting, 28-1
  - Connection Accounting, 28-1
  - Disk Usage, 28-1
- system accounting
  - accounting scripts, 28-3
- system accounting
  - ckpacct, 28-4
- system accounting**
  - Troubleshooting, 28-17**
- System Administrator's Kit, 1-5, 1-7
- system disk, 26-9
- System Load, 3-33**
- system requirements, 2-4
- System Resource statistics, 19-14
- System Status Folder, 19-9
- System Summary information, 19-14
- System\_Admin folder, 3-31
- system\_type, 21-15**

## T

- tacct** file, 28-18
- tail** command, 4-16
- tape drive, 18-7
- tar, 3-17**, **18-2**, 18-10, 18-11, 18-12, **18-14**, 18-15, 18-16, **18-18**, 20-10, **23-11**, **23-12**, **23-22**, **23-23**, 29-14, 30-4
- TCP, 27-2, 27-7
  - Logging, **24-5**
- TCP connections, 24-5

TCP packets, 27-2  
 TCP part, 16-2  
 TCP ports, 16-14, 30-61  
 TCP/IP, 16-2, 27-2  
 TCP/IP model, 16-3, **16-9**, **16-17**  
**Telephone wires, 16-1**  
**telnet**, **3-12**, **12-35**, **16-3**, **16-4**, **16-7**, 16-14, **16-15**, **19-10**, 20-1, 20-2, 21-29, 26-29, **27-6**, 31-20  
**Telnet**, **16-7**, 20-2, 27-2  
**tempfs** file system, 14-6  
 Temporary File System. See tmpfs  
**TERM** variable, 14-8  
 terminal, 1-10  
 terminal connections. See TTY  
 Terminal Icon, 2-6  
 Terminal Server, 1-10  
**TERMINFO**, **17-8**  
 Text Editor, 3-1, 3-6, 3-27, 3-28, 3-29, 12-5, 12-6, **12-7**, 12-8, 12-9, 12-10, 12-11, 12-12, 12-13, *12-14*, *12-15*, *12-16*, *12-17*, 12-19, 12-20, **12-21**, **12-22**, **12-23**, **12-25**, 12-26, 12-28, 12-29, 12-30, 12-31, 12-32, **12-33**, **12-34**, **12-35**  
**The SMC Location Bar, 19-7**  
**Thick coaxial cable, 16-2**  
**Thin coaxial cable, 16-2**  
 Time of day clock, 1-17  
 Time Zone, 2-4  
 TLS, 30-2  
**tmpfs**, **14-8**, 15-2  
 Tools menu choice, 2-6  
 ToolTalk daemon, 3-36  
**touch** command, 3-22, 4-2, **4-4**, 7-33, 12-19, 12-20, 14-4, **24-4**, 24-5  
*track*, 6-3, *13-6*, 13-13, **14-2**, **14-6**, 15-2, 15-5, **19-16**, 26-5, 30-38  
**Tracking, 17-2**  
 tracks, 13-6, 15-2, 15-5, **15-13**, **17-2**, 20-1, **22-2**, **22-3**  
 transactional devices, 26-1  
 transactional volume  
   *logging device*, 26-4  
 transactional volume  
   *master device*, 26-4  
*transactional volumes*, 26-4  
 Transmission Control Protocol/ Internet Protocol. See TCP/IP  
**Transport Layer, 16-4**, **16-9**, 30-35, 30-63  
 Transport Layer Security. See TLS  
 Trash Can, 3-4, 6-9  
**Troubleshooting, 29-1**  
 TTY, 28-1  
 TTY connections, 28-2  
 turnacct, 28-18  
 Turner icon, 19-8, 26-7, 26-28  
**Twisted pair cable, 16-2**

## U

**udf**, **15-1**  
**UDFS**, **3-20**, **3-32**

UDP, 27-7  
**ufs**, **2-36**, **7-32**, **7-33**, **7-36**, **14-9**, **14-10**, 15-1, **21-14**, 26-21, 26-26, 26-30, **26-34**  
 UFS, 21-8, **26-1**  
**UFS (UNIX), 3-20**  
 UFS file system, 7-30, **14-4**, 15-2, 15-4, 15-5, **21-7**, **21-14**, **25-3**, 26-3, **26-18**  
 UFS logging, 15-11  
**ufsboot**, **6-3**  
**ufsdump**, **11-2**, **18-2**, 18-5, 18-6, 18-7, 18-8, 18-9, 18-10, **18-18**  
**ufsrestore**, **18-2**, **18-6**, **18-7**, 18-9, 18-10, **18-18**  
 UID, 7-1, **7-2**, **7-5**, 7-6, 7-7, 7-11, **7-15**, 7-19, 7-27, 11-7, **14-11**, 15-6, 19-15, **22-4**, 22-8, **22-16**, **23-5**, **23-6**, 25-6, 28-1, **28-2**, **28-10**, 28-12, 30-27  
 Ultra 5, 1-2, 1-3, 1-4, **1-20**, *1-26*, 14-10, **14-15**, 21-32  
 Ultra 5 workstations, 1-3  
**umount**, **14-4**, 26-26, 26-30  
**umount** command, **26-19**  
**umount -F**, **3-22**, **18-1**  
**umountall**, **14-9**  
*Unbundled Software Package*, 9-2  
**uncompress**, **18-12**, 18-15  
**uncompress** command, 18-15  
**Unicast Address, 27-3**  
 UNIX, 1-1, 1-5, **3-32**, 4-1, **4-2**, **4-4**, **4-5**, **4-14**, **4-17**, 7-30, **8-6**, 8-7, *12-18*, **13-11**, 14-1, 18-13, **19-4**, 20-1, 20-3, 20-4, 20-10, 30-6, *30-9*, 30-29  
 UNIX File System. See UFS  
**unix.0**, **25-10**  
**unix.x**, **25-6**  
**unset**, **12-8**  
**unset** command, 12-8, 12-9  
 unshare Command, **14-13**  
 unshareall, **14-13**, **14-14**  
**until** Control-Flow command, 12-30  
**unzip**, **18-13**  
 unzip utility, 18-14  
**upgrade**, **21-14**  
**usedisk**, **21-15**  
 User ID, 28-1, See UID  
 User Identification. See UID  
**User Management, 22-4**  
**User Security, 22-4**  
 User Templates, 7-18, **7-21**  
 user\_attr database file, 22-6  
 user's profile files, 8-4  
**useradd**, **2-37**, **7-5**, **7-6**, 7-11, **7-31**, **7-34**, **7-37**, *12-2*, **22-5**, **22-16**, 30-6  
**userdel**, **7-1**, **7-5**, 7-8, 7-9, **7-37**  
**usermod**, **7-1**, **7-5**, 7-8, **7-37**  
**users.deny**, **17-8**

## V

Value Added Resellers. See VAR  
 VAR, 1-4

var/named/loopback-domain-info, 29-16  
var/named/named.run, 29-18  
variable length subnet mask, 31-15  
Variables, 12-6  
Veritas Cluster Server, 26-3  
**vfgetprojnet**, 19-16  
**vi**, xvi, 1-8, 3-27, 3-29, 4-1, 4-2, 4-4, 4-5, 4-6, 4-7, 4-8, 4-9, 4-10, 4-11, 4-12, 4-13, 4-14, 4-15, 4-16, 4-17, 5-6, 6-8, 11-3, 12-5, 14-8, 16-17, 21-18, 21-19, 21-26, 21-27, 21-31, 23-9, 23-10, 23-19, 23-20, 26-26, 28-7, 28-8, 28-14, 28-19, 29-19, 30-22, 30-28, 30-50, 30-55  
Vi move the cursor, 4-6, 4-13  
video adapter. *See* 13W3  
view floppies, 3-21  
virtual disks, 26-3  
**vmcore.0**, 25-10  
**vmcore.x**, 25-6  
Voice over IP. *See* VOIP  
VOIP, 27-1  
**volcheck**, 3-21, 3-22, 14-5, 18-1, 18-2, 18-8, 18-10, 21-17, 21-19  
**void** command, 14-5  
**volmgt**, 18-1  
**volmgt stop**, 3-22, 14-5, 18-1, 18-8  
*Volume*, 26-2  
Volume Name, 26-23  
*Volume Table of Contents*. *See* VTOC  
Volumes, 26-2  
**vt100**, 14-8, 17-5  
VT100, 1-15  
*VTOC*, 15-3

## W

**WAN**, 16-1  
WBEM logging service, 19-13  
**wc**, 12-14  
**Web Start Flash**, 2-2, 20-6, 21-2, 21-14  
Web Start installation, 2-1, 2-2, 2-7, 21-1  
Web Start Installation, 2-1, 2-11, 2-20, 9-11  
**Web Start Installer**, 9-1, 9-11, 9-14  
Web Start program, 9-11  
**which** utility, 8-3  
**while** Control-Flow command, 12-27  
**whois**, 16-14  
**Wide Area Network**. *See* WAN  
*word count*. *See* wc  
Workspace button, 3-3, 3-6, 3-7, 3-8  
Workspace buttons, 3-3, 3-6, 3-7, 3-8

Workspace Manager Controls icon, 3-8  
Workspace menu, 2-5, 2-39, 3-2, 3-9, 3-10, 3-11, 3-16, 4-3, 4-4, 6-8, 7-11, 10-6, 11-4, 12-5, 14-2, 16-11, 19-1, 22-8, 24-3, 26-6, 26-9, 26-16, 28-3  
workspaces, 3-2, 3-3, 3-6, 3-7, 20-7  
**wq**, 4-6  
write permission, 7-9  
**wtmperror**, 28-7  
**wtmpx**, 28-19

## X

X server configuration files, 3-9  
X.500 directories, 30-3  
X.500 directory, 30-3  
XBM format, 3-33  
X-chat, 3-2, 3-7  
XDR, 14-10  
XPM file format, 3-33

## Y

Yes Tester, 12-22  
**ypbind**, 23-12  
**ypcat**, 23-5  
**ypinit**, 23-6, 23-8  
**ypmake**, 23-5  
**yppoll**, 23-6  
**yppush**, 23-6  
**ypserv**, 23-12  
**ypset**, 23-6  
**ypstart**, 23-6  
**ypstop**, 23-6  
**ypwhich**, 23-6, 23-10  
**ypxfrd**, 23-12

## Z

**zcat**, 18-12  
**zcat** utility, 18-15  
**zip**, 18-12, 23-11  
**zip** file, 18-13  
**zip** program, 18-13  
**zip** utility, 18-13  
**Zone Master Server**, 29-1  
**Zone Slave Server**, 29-1  
**Zone Stub Server**, 29-1