Oracle® Solaris Administration: Common Tasks

Part No: 821–1451–11
December 2011

ORACLE

111206@25097

Contents

Preface 17

1 Locating Information About Oracle Solaris Commands 23
Support for Full-Text Searching of Man Pages 23
About Oracle Solaris Man Pages 24
Locating Information in Man Pages 24
Creating Index Files to Enable the Searching of Man Pages for Specific Keywords 24
Searching for Information in Man Pages by Specifying Keywords 25
Format of the Man Page Sections 29
Description of the Man Page Sections 29
Man Page Format 29

2 Managing User Accounts and Groups (Overview) 33
What's New in User Accounts and Groups? 33
Removal of Support for Solaris Management Console 33
Default Password Hashing Algorithm 33
What Are User Accounts and Groups? 34
User Account Components 34
Guidelines for Assigning User Names, User IDs, and Group IDs 40
Where User Account and Group Information Is Stored 41
Fields in the `passwd` File 41
Default `passwd` File 42
Fields in the `shadow` File 44
Fields in the `group` File 44
Default `group` File 44
Command-Line Tools for User and Group Account Management 46
Customizing a User's Work Environment 47

Using Site Initialization Files 48
Avoiding Local System References 48
Shell Features 49
Bash and ksh93 Shell History 49
Bash and ksh93 Shell Environment Variables 50
Customizing the Bash Shell 53
About the MANPATH Environment Variable 53
The PATH Variable 53
Locale Variables 54
Default File Permissions (umask) 55
Customizing a User Initialization File 56

3 Managing User Accounts and Groups (Tasks) 57
Setting Up and Administering User Accounts (Task Map) 57
Setting Up User Accounts 58
Gathering User Information 59
▼ How to Customize User Initialization Files 60
▼ How to Change Account Defaults For All Roles 60
▼ How to Add a User 61
▼ How to Delete a User 62
▼ How to Add a Group 63
▼ How to Share Home Directories That Are Created as ZFS File Systems 63
Manually Mounting a User's Home Directory. 64

4 Booting and Shutting Down an Oracle Solaris System 67
What's New in Booting and Shutting Down a System? 67
Support for Administratively Provided driver.conf Files 68
Bitmapped Console 69
Boot and Shutdown Progress Animation 69
Fast Reboot 69
x86: Removal of Support for 32–Bit Kernel 70
Booting and Shutting Down an Oracle Solaris System (Overview) 70
GRUB Based Booting 71
Management of Boot Services by the Service Management Facility 71
Booting a System to a Specified State (Task Map) 72

Booting a System to a Specified State (Run Level) 72
Determining a System's Current Run Level 73
▼ SPARC: How to Boot a System to a Multiuser State (Run Level 3) 73
▼ x86: How to Boot a System to a Single-User State (Run Level S) 74
Shutting Down a System (Task Map) 76
Shutting Down a System 76
▼ How to Shut Down a System by Using the `shutdown` Command 76
Bringing a System to a Shutdown State (Run Level 0) by Using the `init` Command 78
▼ How to Shut Down a System by Using the `init` Command 78
Booting a System From the Network 79
Accelerating the Reboot Process (Task Map) 80
Accelerating the Reboot Process 81
▼ How to Initiate a Fast Reboot of a SPARC Based System 81
▼ How to Initiate a Fast Reboot of an x86 Based System 82
Changing the Default Behavior of the Fast Reboot Feature 82
Initiating a Standard Reboot of a System That Has Fast Reboot Enabled 83
Booting From a ZFS Boot Environment (Task Map) 83
SPARC: Booting From a ZFS Boot Environment 84
▼ SPARC: How to Display a List of Available Boot Environments During the Boot Sequence 85
▼ SPARC: How to Boot From a ZFS Boot Environment or Root File System 85
Modifying Boot Parameters (Task Map) 87
Modifying Boot Parameters 88
▼ SPARC: How to Determine the Default Boot Device 89
▼ SPARC: How to Change the Default Boot Device by Using the Boot PROM 90
▼ x86: How to Modify Boot Parameters by Using the `eeprom` Command 91
▼ x86: How to Modify Boot Parameters at Boot Time 92
Adding a Linux Entry to the GRUB Menu After an Installation 93
Keeping a System Bootable (Task Map) 93
Keeping a System Bootable 94
Determining Whether the `boot-archive` SMF Service Is Running 94
▼ How to Clear a Failed Automatic Boot Archive Update by Manually Updating the Boot Archive 95
▼ x86: How to Clear a Failed Automatic Boot Archive Update by Using the `auto-reboot-safe` Property 95
Where to Find More Information About Booting and Shutting Down a System 96

5 Working With Oracle Configuration Manager 99
Introduction to Oracle Configuration Manager 99
Managing Oracle Configuration Manager (Tasks) 100
▼ How to Enable the Oracle Configuration Manager Service 101
▼ How to Disable the Oracle Configuration Manager Service 101
▼ How to Manually Register With the Oracle Repository 101
▼ How to Change the Time or Frequency of Data Collection 102

6 Managing Services (Overview) 103
Introduction to SMF 103
SMF Concepts 104
SMF Service 104
SMF Dependencies 105
Service Identifiers 105
Service States 106
SMF Manifests 106
SMF Profiles 107
Service Configuration Repository 107
SMF Repository Backups 108
SMF Snapshots 109
SMF Administrative Layers 109
SMF Service Error Logging 110
SMF Administrative and Programming Interfaces 110
SMF Command-Line Administrative Utilities 110
Service Management Configuration Library Interfaces 111
SMF Components 111
SMF Master Restarter Daemon 111
SMF Delegated Restarters 112
SMF Properties and Property Groups 112
Managing Information in the Service Configuration Repository 113
Viewing SMF Information 113
Modifying SMF Information 113
Deleting SMF Information 114
SMF and Booting 114
SMF Compatibility 115

Run Levels 115
When to Use Run Levels or Milestones 116
Determining a System's Run Level 116
/etc/inittab File 117
What Happens When the System Is Brought to Run Level 3 118

7 Managing Services (Tasks) 119
Monitoring Services (Task Map) 119
Monitoring SMF Services 120
▼ How to List the Status of a Service 120
▼ How to List Customizations of a Service 121
▼ How to Show Which Services Are Dependent on a Service Instance 121
▼ How to Show Which Services a Service Is Dependent On 122
▼ How to Set Up Email Notification of SMF Transition Events 122
Managing SMF Services (Task Map) 124
Managing SMF Services 125
Using RBAC Rights Profiles With SMF 125
▼ How to Disable a Service Instance 125
▼ How to Enable a Service Instance 125
▼ How to Restart a Service 126
▼ How to Restore a Service That Is in the Maintenance State 127
▼ How to Create an SMF Profile 127
▼ How to Apply an SMF Profile 129
Configuring SMF Services (Task Map) 129
Configuring SMF Services 129
▼ How to Modify an SMF Service Property 130
▼ How to Modify a Service That Is Configured by a File 130
▼ How to Change an Environment Variable for a Service 131
▼ How to Change a Property for an inetd Controlled Service 131
▼ How to Delete Customizations for a Service 133
▼ How to Modify a Command-Line Argument for an inetd Controlled Service 133
▼ How to Convert inetd.conf Entries 134
Using Run Control Scripts (Task Map) 135
Using Run Control Scripts 135
▼ How to Use a Run Control Script to Stop or Start a Legacy Service 135

▼ How to Add a Run Control Script 136
▼ How to Disable a Run Control Script 137
Troubleshooting the Service Management Facility 138
▼ Debugging a Service That Is Not Starting 138
▼ How to Repair a Corrupt Repository 138
▼ How to Boot Without Starting Any Services 141
▼ How to Force an `sulogin` Prompt If the `system/filesystem/local:default` Service Fails During Boot 141

8 Using the Fault Manager 143
Fault Management Overview 143
Notification of Faults and Defects 145
Displaying Information About Faults or Defects 145
▼ How to Display Information About Faulty Components 146
▼ How to Identify Which CPUs Are Offline 148
▼ How to Display Information About Defective Services 148
Repairing Faults or Defects 149
`fmadm replaced` Command 150
`fmadm repaired` Command 150
`fmadm acquit` Command 150
Fault Management Log Files 151
Fault Statistics 151

9 Managing System Information (Tasks) 153
What's New in Displaying and Changing System Information 153
Support for Administratively Provided `driver.conf` Files 153
Displaying System Information (Task Map) 154
Displaying System Information 155
▼ How to Display a System's Release Information 155
▼ How to Display a System's Host ID Number 156
▼ How to Display a System's Product Name 156
▼ How to Display a System's Installed Memory 156
▼ How to Display Default and Customized Property Values for a Device 157
▼ How to Display the Date and Time 158
Identifying Information About Chip Multithreading Features 158

▼ How to Display a System's Physical Processor Type 158
▼ How to Display a System's Logical Processor Type 159
Changing System Information (Task Map) 160
Changing System Information 160
▼ How to Manually Set a System's Date and Time 160
▼ How to Set Up a Message-Of-The-Day 161
▼ How to Change a System's Identity (`nodename`) 162

10 Managing System Processes (Tasks) 163
Managing System Processes (Task Map) 163
Commands for Managing System Processes 164
Using the `ps` Command 164
Using the `/proc` File System and Commands 165
Managing Processes With Process Commands (`/proc`) 166
▼ How to List Processes 167
▼ How to Display Information About Processes 168
▼ How to Control Processes 169
Terminating a Process (`pkill, kill`) 170
▼ How to Terminate a Process (`pkill`) 170
▼ How to Terminate a Process (`kill`) 171
Debugging a Process (`pargs`, `preap`) 172
Managing Process Class Information (Task Map) 173
Managing Process Class Information 173
Changing the Scheduling Priority of Processes (`priocntl`) 174
▼ How to Display Basic Information About Process Classes (`priocntl`) 174
▼ How to Display the Global Priority of a Process 175
▼ How to Designate a Process Priority (`priocntl`) 176
▼ How to Change Scheduling Parameters of a Timesharing Process (`priocntl`) 177
▼ How to Change the Class of a Process (`priocntl`) 178
Changing the Priority of a Timesharing Process (`nice`) 178
▼ How to Change the Priority of a Process (`nice`) 179
Troubleshooting Problems With System Processes 180

11 Monitoring System Performance (Tasks) 181
Where to Find System Performance Tasks 181

System Performance and System Resources 182
Processes and System Performance 182
About Monitoring System Performance 184
Monitoring Tools 184
Displaying System Performance Information (Task Map) 185
Displaying Virtual Memory Statistics (vmstat) 186
▼ How to Display Virtual Memory Statistics (vmstat) 187
▼ How to Display System Event Information (vmstat -s) 188
▼ How to Display Swapping Statistics (vmstat -S) 188
▼ How to Display Interrupts Per Device (vmstat -i) 189
Displaying Disk Utilization Information (iostat) 189
▼ How to Display Disk Utilization Information (iostat) 190
▼ How to Display Extended Disk Statistics (iostat -xtc) 191
Displaying Disk Space Statistics (df) 192
▼ How to Display Disk Space Information (df -k) 192
Monitoring System Activities (Task Map) 193
Monitoring System Activities (sar) 195
▼ How to Check File Access (sar -a) 195
▼ How to Check Buffer Activity (sar -b) 196
▼ How to Check System Call Statistics (sar -c) 197
▼ How to Check Disk Activity (sar -d) 199
▼ How to Check Page-Out and Memory (sar -g) 200
Checking Kernel Memory Allocation 201
▼ How to Check Kernel Memory Allocation (sar -k) 202
▼ How to Check Interprocess Communication (sar -m) 203
▼ How to Check Page-In Activity (sar -p) 204
▼ How to Check Queue Activity (sar -q) 205
▼ How to Check Unused Memory (sar -r) 206
▼ How to Check CPU Utilization (sar -u) 207
▼ How to Check System Table Status (sar -v) 208
▼ How to Check Swapping Activity (sar -w) 209
▼ How to Check Terminal Activity (sar -y) 210
▼ How to Check Overall System Performance (sar -A) 211
Collecting System Activity Data Automatically (sar) 211
Running the sadc Command When Booting 212
Running the sadc Command Periodically With the sa1 Script 212

Producing Reports With the sa2 Shell Script 212
Setting Up Automatic Data Collection (sar) 213
▼ How to Set Up Automatic Data Collection 214

12 Managing Software Packages (Tasks) 215
Managing Software Packages (Task Map) 215
Image Packaging System 216
Getting Information About Packages 216
Installing and Updating Packages 219
Installing a New Package 219
Updating All Installed Packages 222

13 Managing Disk Use (Tasks) 223
Managing Disk Use (Task Map) 223
Displaying Information About Files and Disk Space 224
▼ How to Display Information About Files and Disk Space 225
Checking the Size of Files 227
▼ How to Display the Size of Files 227
▼ How to Find Large Files 228
▼ How to Find Files That Exceed a Specified Size Limit 229
Checking the Size of Directories 230
▼ How to Display the Size of Directories, Subdirectories, and Files 230
Finding and Removing Old or Inactive Files 231
▼ How to List the Newest Files 231
▼ How to Find and Remove Old or Inactive Files 232
▼ How to Clear Out Temporary Directories 233
▼ How to Find and Delete core Files 234
▼ How to Delete Crash Dump Files 234

14 Scheduling System Tasks (Tasks) 237
Creating and Editing crontab Files (Task Map) 237
Ways to Automatically Execute System Tasks 238
For Scheduling Repetitive Jobs: crontab 238
For Scheduling a Single Job: at 239

Scheduling a Repetitive System Task (cron) 240
Inside a crontab File 240
How the cron Daemon Handles Scheduling 241
Syntax of crontab File Entries 241
Creating and Editing crontab Files 242
▼ How to Create or Edit a crontab File 243
▼ How to Verify That a crontab File Exists 244
Displaying crontab Files 244
▼ How to Display a crontab File 244
Removing crontab Files 245
▼ How to Remove a crontab File 245
Controlling Access to the crontab Command 246
▼ How to Deny crontab Command Access 247
▼ How to Limit crontab Command Access to Specified Users 248
How to Verify Limited crontab Command Access 249
Using the at Command (Task Map) 249
Scheduling a Single System Task (at) 250
Description of the at Command 250
Controlling Access to the at Command 251
▼ How to Create an at Job 251
▼ How to Display the at Queue 252
▼ How to Verify an at Job 252
▼ How to Display at Jobs 252
▼ How to Remove at Jobs 253
▼ How to Deny Access to the at Command 254
▼ How to Verify That at Command Access Is Denied 255

15 Setting Up and Administering Printers by Using CUPS (Tasks) 257
Introduction to CUPS 257
CUPS Processes 258
CUPS Services 258
Setting Up Printers and Print Queues by Using CUPS 259
Managing Print Requests by Using CUPS 259
Setting Up Your Printing Environment to Work With CUPS 260
▼ How to Set Up Your Printing Environment 261

Setting Up Your Printing Environment for an Upgrade 262
Setting Up and Administering Printers by Using CUPS Command-Line Utilities (Task Map) 262
Setting Up and Administering Printers by Using CUPS Command-Line Utilities 263
CUPS Command-Line Utilities 263
▼ How to Set Up a Printer by Using the `lpadmin` Command 264
Setting a Default Printer 265
▼ How to Verify the Status of Printers 268
▼ How to Print a File to the Default Printer 269
▼ How to Delete a Printer and Remove Printer Access 270
Setting Up and Administering Printers by Using the CUPS Web Browser Interface (Task Map) 271
Setting Up and Administering Printers by Using the CUPS Web Browser Interface 272
Requirements for Using the CUPS Web Browser Interface 272
Troubleshooting Issues With Accessing the CUPS Web Browser Interface 273
Print Administration Tasks 273
About the Administration Tab 274
About the Printers Tab 275
▼ How to Add a New Printer 276
About the CUPS Print Manager GUI 276
Starting CUPS Print Manager 276
Setting Up Printers by Using CUPS Print Manager (Task Map) 277
Setting Up Printers by Using CUPS Print Manager 278
Local Server Configuration 278
Remote Server Configuration 278
Selecting a Print Device 280
▼ How to Set Up a New Local Printer 281
Administering Printers by Using CUPS Print Manager (Task Map) 283
Administering Printers by Using CUPS Print Manager 283
Configurable Printer Properties 283
▼ How to Modify the Properties of a Configured Printer 285
▼ How to Rename a Printer 286
▼ How to Copy a Printer Configuration 287
▼ How to Delete a Printer 287
▼ How to Unshare or Share a Printer 287
▼ How to Disable or Enable a Printer 288

▼ How to Manage Print Jobs for a Specified Printer 288

16 Managing the System Console, Terminal Devices, and Power Services (Tasks) 291

What's New in Managing the System Console and Locally Connected Terminal Devices 291

Removal of Support for SVR4 Service Access Facility Commands and Service Access Controller Program 291

Virtual Terminal Support 292

Bitmapped Console Support 292

Managing the System Console and Locally Attached Connected Terminal Devices (Task Map) 293

Overview of the System Console and Locally Connected Terminal Devices 293

SMF Services That Manage the System Console and Locally Connected Terminal Devices 293

Managing the System Console and Locally Connected Terminal Devices 294

▼ How to Modify Settings for the System Console 294

▼ How to Set Up Login Services on Auxiliary Terminals 295

▼ How to Set the Baud Rate Speed on the System Terminal 295

Managing System Power Services 296

▼ How to Recover from Power Service in Maintenance Mode 297

17 Managing System Crash Information (Tasks) 299

What's New in Managing System Crash Information 299

Fast Crash Dump Facility 299

Managing System Crash Information (Task Map) 300

System Crashes (Overview) 300

x86: System Crashes in the GRUB Boot Environment 301

System Crash Dump Files 301

Saving Crash Dumps 301

The `dumpadm` Command 302

How the `dumpadm` Command Works 303

Managing System Crash Dump Information 303

▼ How to Display the Current Crash Dump Configuration 303

▼ How to Modify a Crash Dump Configuration 304

▼ How to Examine a Crash Dump 305

▼ How to Recover From a Full Crash Dump Directory (Optional) 306

▼ How to Disable or Enable the Saving of Crash Dumps 307

18 Managing Core Files (Tasks) 309
Managing Core Files (Task Map) 309
Managing Core Files Overview 310
Configurable Core File Paths 310
Expanded Core File Names 310
Setting the Core File Name Pattern 311
Enabling `setuid` Programs to Produce Core Files 312
How to Display the Current Core Dump Configuration 312
▼ How to Set a Core File Name Pattern 313
▼ How to Enable a Per-Process Core File Path 313
▼ How to Enable a Global Core File Path 313
Troubleshooting Core File Problems 314
Examining Core Files 314

19 Troubleshooting System and Software Problems (Tasks) 315
Troubleshooting a System Crash 315
What to Do If the System Crashes 315
Gathering Troubleshooting Data 316
Troubleshooting a System Crash Checklist 317
Managing System Messages 317
Viewing System Messages 318
System Log Rotation 319
Customizing System Message Logging 320
Enabling Remote Console Messaging 322
Troubleshooting File Access Problems 326
Solving Problems With Search Paths (`Command not found`) 327
Changing File and Group Ownerships 328
Solving File Access Problems 328
Recognizing Problems With Network Access 329

20 Troubleshooting Miscellaneous System and Software Problems (Tasks) 331
What to Do If Rebooting Fails 331

What to Do If You Forgot the Root Password or Problem That Prevents System From Booting 332
What to Do If a System Hang Occurs 333
What to Do If a File System Fills Up 333
File System Fills Up Because a Large File or Directory Was Created 334
A TMPFS File System Is Full Because the System Ran Out of Memory 334
What to Do If File ACLs Are Lost After Copy or Restore 334

Index 335

Preface

System Administration Guide: Common System Management Tasks is part of a documentation set that provides a significant portion of the Oracle Solaris system administration information. This guide contains information for both SPARC based and x86 based systems.

This book assumes you have completed the following tasks:

- Installed the Oracle Solaris 11 software
- Set up all the networking software that you plan to use

For the Oracle Solaris 11 release, new features that might be interesting to system administrators are covered in sections called *What's New in ... ?* in the appropriate chapters.

Note – This Oracle Solaris release supports systems that use the SPARC and x86 families of processor architectures. The supported systems appear in the *Oracle Solaris OS: Hardware Compatibility Lists*. This document cites any implementation differences between the platform types.

For supported systems, see the *Oracle Solaris OS: Hardware Compatibility Lists*.

Who Should Use This Book

This book is intended for anyone responsible for administering one or more systems running the Oracle Solaris 11 release. To use this book, you should have 1–2 years of UNIX system administration experience. Attending UNIX system administration training courses might be helpful.

How the System Administration Guides Are Organized

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

Book Title	Topics
Booting and Shutting Down Oracle Solaris on SPARC Platforms	Booting and shutting down a system, managing boot services, modifying boot behavior, booting from ZFS, managing the boot archive, and troubleshooting booting on SPARC platforms
Booting and Shutting Down Oracle Solaris on x86 Platforms	Booting and shutting down a system, managing boot services, modifying boot behavior, booting from ZFS, managing the boot archive, and troubleshooting booting on x86 platforms
Oracle Solaris Administration: Common Tasks	Using Oracle Solaris commands, booting and shutting down a system, managing user accounts and groups, managing services, hardware faults, system information, system resources, and system performance, managing software, printing, the console and terminals, and troubleshooting system and software problems
Oracle Solaris Administration: Devices and File Systems	Removable media, disks and devices, file systems, and backing up and restoring data
Oracle Solaris Administration: IP Services	TCP/IP network administration, IPv4 and IPv6 address administration, DHCP, IPsec, IKE, IP Filter, and IPQoS
Oracle Solaris Administration: Naming and Directory Services	DNS, NIS, and LDAP naming and directory services, including transitioning from NIS to LDAP
Oracle Solaris Administration: Network Interfaces and Network Virtualization	Automatic and manual IP interface configuration including WiFi wireless; administration of bridges, VLANs, aggregations, LLDP, and IPMP; virtual NICs and resource management.
Oracle Solaris Administration: Network Services	Web cache servers, time-related services, network file systems (NFS and autofs), mail, SLP, and PPP
Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management	Resource management features, which enable you to control how applications use available system resources; Oracle Solaris Zones software partitioning technology, which virtualizes operating system services to create an isolated environment for running applications; and Oracle Solaris 10 Zones, which host Oracle Solaris 10 environments running on the Oracle Solaris 11 kernel
Oracle Solaris Administration: Security Services	Auditing, device management, file security, BART, Kerberos services, PAM, Cryptographic Framework, Key Management Framework, privileges, RBAC, SASL, Secure Shell and virus scanning.
Oracle Solaris Administration: SMB and Windows Interoperability	SMB service, which enables you to configure an Oracle Solaris system to make SMB shares available to SMB clients; SMB client, which enables you to access SMB shares; and native identity mapping service, which enables you to map user and group identities between Oracle Solaris systems and Windows systems

Book Title	Topics
Oracle Solaris Administration: ZFS File Systems	ZFS storage pool and file system creation and management, snapshots, clones, backups, using access control lists (ACLs) to protect ZFS files, using ZFS on an Oracle Solaris system with zones installed, emulated volumes, and troubleshooting and data recovery
Trusted Extensions Configuration and Administration	System installation, configuration, and administration that is specific to Trusted Extensions
Oracle Solaris 11 Security Guidelines	Securing an Oracle Solaris system, as well as usage scenarios for its security features, such as zones, ZFS, and Trusted Extensions
Transitioning From Oracle Solaris 10 to Oracle Solaris 11	Provides system administration information and examples for transitioning from Oracle Solaris 10 to Oracle Solaris 11 in the areas of installation, device, disk, and file system management, software management, networking, system management, security, virtualization, desktop features, user account management, and user environments emulated volumes, and troubleshooting and data recovery

Related Third-Party Web Site References

Note – Oracle is not responsible for the availability of third-party web sites mentioned in this document. Oracle does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Oracle will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Access to Oracle Support

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

Typographic Conventions

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

TABLE P–1 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`
aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX system prompt and superuser prompt for shells that are included in the Oracle Solaris OS. Note that the default system prompt that is displayed in command examples varies, depending on the Oracle Solaris release.

TABLE P–2 Shell Prompts

Shell	Prompt
Bash shell, Korn shell, and Bourne shell	`$`
Bash shell, Korn shell, and Bourne shell for superuser	`#`
C shell	`machine_name%`
C shell for superuser	`machine_name#`

General Conventions

Be aware of the following conventions used in this book.

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

◆ ◆ ◆ CHAPTER 1

Locating Information About Oracle Solaris Commands

This chapter provides information about locating information about Oracle Solaris commands, specifically how to search man (short for "reference manual") pages for information about commands.

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

- "Support for Full-Text Searching of Man Pages" on page 23
- "About Oracle Solaris Man Pages" on page 24
- "Locating Information in Man Pages" on page 24
- "Format of the Man Page Sections" on page 29

Support for Full-Text Searching of Man Pages

To support full-text searching of man pages, the `man` command has been enhanced to include a new `-K` (uppercase) option that enables you to search for specific keywords throughout all of the sections in all of the man pages. The `-K` option works similarly to the `-k` (lowercase) option, with the exception that the `-k` option is limited to searching only the `NAME` subsection of all of the man page sections. The new `-K` option is useful for locating information about how a command is used in a variety of contexts.

The `-k` and `-K` options of the `man` command utilize index files for searching. To support the autogeneration of new index files as man pages are updated, a new Service Management Facility (SMF) feature of Oracle Solaris service has been introduced. The `svc:/application/man-index:default` service triggers automatic regeneration of new index files whenever new man pages are added to the `/usr/share/man` and `/usr/gnu/share/man` directories if these directories exist. This service is enabled by default.

Note that for alternate man page directories, such as `/opt/SUNWspro/man`, you can create a symbolic link within the `/usr/share/man/index.d` directory to point to the alternate man page directory.

For instructions and examples, see "Searching for Information in Man Pages by Specifying Keywords" on page 25.

About Oracle Solaris Man Pages

Oracle Solaris includes extensive reference materials that are known as *man pages*. Each page is a self-contained document that describes one or more UNIX constructs. A man page could describe a command, library function, file format, or a device driver. For virtually every Oracle Solaris command, a man page is provided. The collective man pages, organized alphabetically and by sections, comprise a reference manual.

A man page is intended to answer concisely the question, "What does the command (or construct) do?" A man page is not a tutorial or a technical analysis of Oracle Solaris internals. Also, man pages do not document detailed procedures. However, man pages do provide examples of command usage.

To display a man page, type the command `man` *command-name* in a terminal window. For example, to display the man page for the `boot` command, you would type the following:

```
$ man boot
Reformatting page.  Please Wait... done  System Administration Commands boot(1M)
NAME      boot - start the system kernel or a standalone program
.
.
.
```

Locating Information in Man Pages

You can search for information about Oracle Solaris commands in man pages by using the `man` command. The new `-K` (uppercase) *keywords* option enables you search all sections of all of the man pages for the specified keywords. The `-k` option (lowercase) is limited to searching just the `NAME` subsection of all of the man page sections.

Creating Index Files to Enable the Searching of Man Pages for Specific Keywords

To search for information in man pages by specifying keywords, start by creating index files that can be queried by the `man` command. You can use the `catman` utility to create preformatted versions of the online reference manual. When used without any options, the `catman` utility generates index files of the man pages that are in the `/usr/share/man` and `usr/gnu/share/man` directories. Note that you must be the `root` user to use the `catman` utility.

The `catman` utility also includes a `-w` option. This option enables you to create an index file of man pages in the directories that are specified by the `MANPATH` variable or when the `-M` option is

used. You can use the `-M` option of the `catman` utility to specify an alternate directory to create an index file of third-party man pages. By default, when used without any options, the `catman` utility creates an index file of the man pages that are located in the `/usr/share/man` and `/usr/gnu/share/man` directories.

To specify an alternate directory, type the following command:

```
# catman -M directory -w
```

For example, to create an index file for man pages that are located in the `/usr/local/share/man` directory, you would type the following command:

```
# catman -M /usr/local/share/man -w
```

`-M` *directory* — Updates the man pages that are located in the specified directory, which is `/usr/share/man` by default. If the `-M` option is specified, the directory argument must not contain a comma (`,`) because a comma is used to delineate section numbers in man pages.

`-w` — Creates an index file in the directories that are specified by the `MANPATH` variable or with the `-M` option. If the `MANPATH` variable or the `-M` option is not specified, index files are created for the both the `/usr/share/man/` and `/usr/gnu/share/man` directories.

For a complete description of the `catman` utility, including all of the command-line options that you can specify, see the `catman(1M)` man page.

Searching for Information in Man Pages by Specifying Keywords

To search for information in man pages by specifying keywords, create an index file, or files, by using the `catman` utility. For instructions, see "Creating Index Files to Enable the Searching of Man Pages for Specific Keywords" on page 24.

To conduct a full-text search for information in all of the man page sections by specifying keywords, type the `man` command with the new `-K` *keywords* option, as follows:

```
$ man -K keywords
```

`-K` — Conducts a full-text search of the specified *keywords* throughout all of the man page sections.

keywords — Is a user-defined string that is specified. To specify multiple keywords in single search, use double quotation marks. For example:

```
$ man -K "create zfs pool"
```

To search for information in *just* the `NAME` subsection of all of the man pages, type the `man` command with the `-k` *keywords* option, as follows:

```
$ man -k keywords
```

To search for information in a specific subsection of all of the man pages, include the *SECTNAME* in the `man -K` *keywords* command syntax, as follows:

```
$ man -K SECTNAME:keywords
```

where *SECTNAME* can be any of the following subsections in all of the man page sections:

- `ATTRIBUTES`
- `DESCRIPTION`
- `ENVIRONMENT VARIABLES`
- `EXAMPLES`
- `EXIT STATUS`
- `FILES`
- `LIST OF COMMANDS`
- `NAME`
- `NOTES`
- `NULL`
- `OPTIONS`
- `OPERANDS`
- `OUTPUT`
- `SECURITY`
- `SEE ALSO`
- `SYNOPSIS`
- `USAGE`

To search a specific man page section, use the `-s` option with the `man` command and the `-k` or `-K` option.

```
$ man -s manpage-section -K "keywords"
```

For example, to search for the keywords “cpu usage” in the 1M man page section, you would type the following command:

```
$ man -s 1m -K "cpu usage"
```

Note – Keywords are contained within double quotation marks.

Note the following additional key information about the `man` command:

- The command syntax that is used for searching man pages by using the `man` command with either the `-k` or `-K` option is not case-sensitive.
- The `man` command normalizes keywords before conducting a search, which means a query for a specified keyword also generates results for all words that include part of that word. For example, if you query the word "searching," the words "search," "searches," and "searched," are also included in the search results.

EXAMPLE 1–1 Searching for Commands in the `NAME` Subsection of All Man Pages by Specifying Keywords

In the following example, the `man -k` command is used to search for instances of the `init` command in the `NAME` section of all man pages *only*:

```
$ man -k init
Searching in: /usr/man
Searching in: /usr/dt/man
Searching in: /usr/openwin/share/man
Searching in: /usr/sfw/man

1. init(1m)     "NAME"  /usr/share/man/man1m/init.1m
init, telinit - process control initialization


2. inittab(4)   "NAME"  /usr/share/man/man4/inittab.4
inittab - script for init


3. init.d(4)    "NAME"  /usr/share/man/man4/init.d.4
init.d - initialization and termination scripts for changing init states
```

In this example, the output of the `man -k` command displays the search results for the `init` command in the `NAME` subsection of all man page sections only. Note that any man page that contains `init` in its name is also included in the search results.

EXAMPLE 1–2 Searching for Commands in All Man Page Sections by Specifying Keywords

In the following example, the `man -K` *keywords* command is used to search for instances of the `dumpadm` command in all of the sections of the man pages.

```
$ man -K dumpadm
Searching in: /usr/man
Searching in: /usr/dt/man
Searching in: /usr/openwin/share/man
Searching in: /usr/sfw/man

1. dumpadm(1m)  NAME    /usr/share/man/man1m/dumpadm.1m
dumpadm - configure operating system crash dump


2. savecore(1m) DESCRIPTION     /usr/share/man/man1m/savecore.1m
The savecore utility saves a crash dump of the kernel (assuming that one
was made) and writes a reboot message in the shutdown log. It is invoked
by the dumpadm service each time the system boots.
```

EXAMPLE 1–2 Searching for Commands in All Man Page Sections by Specifying Keywords *(Continued)*

```
3. svccfg(1m)   EXAMPLES        /usr/sh
```

In this example, the search returns three instances of the `dumpadm` command in the man pages:

- The first instance was found in the `NAME` subsection of the `dumpadm(1M)` man page.
- The second instance was found in the `DESCRIPTION` subsection of the `savecore(1M)` man page.
- The third instance was found in the `EXAMPLES` subsection of the `svccfg(1M)` man page.

Based on the results of this query, the user knows where additional information about the `dumpadm` command can be found in all of the man pages.

EXAMPLE 1–3 Searching for Commands in Specific Man Page Subsections by Specifying Keywords

In the following example, the `man -K` command is used with the *SECTNAME*:*keywords* argument to search for instances of the `bootadm` command in the `DESCRIPTION` subsection of all of the man page sections.

```
$ man -K description:bootadm
Searching in: /usr/man
Searching in: /usr/dt/man
Searching in: /usr/openwin/share/man
Searching in: /usr/sfw/man

1. bootadm(1m)  DESCRIPTION     /usr/share/man/man1m/bootadm.1m
The bootadm command manages the boot archive and, with x86 boot environments,
the GRUB (GRand Unified Bootloader) menu. The update-archive option
provides a way for user to update the boot archive as a preventative
measure or as part of a


2. boot(1m)     DESCRIPTION     /usr/share/man/man1m/boot.1m
# bootadm update-archive
```

In this example, the search returns two instances of the `bootadm` command in the `DESCRIPTION` subsection of the following man page sections:

- The first instance was found in the `bootadm(1M)` man page.
- The second instance was found in the `boot(1M)` man page.

For more information about using the `man` command, see the `man(1)` man page.

Format of the Man Page Sections

The following reference information describes the contents of each man page section, the information that is in each section, and the standard layout that is used for all Oracle Solaris man pages.

Description of the Man Page Sections

The following table describes each man page section and the information that each man page references.

TABLE 1–1 Description of Man Page Sections

Man Page Section	Description
1	Describes general commands that are available with the operating system.
1M	Describes commands and daemons that are primarily used for system maintenance and administrative purposes.
2	Describes all of the system calls.
3	Describes functions that are found in various libraries.
4	Outlines the various file formats and conventions.
5	Contains miscellaneous documentation, such as standards, environments, and macros.
6	Contains games and demos. Note that this section no longer exists.
7	Describes special files that refer to specific hardware peripherals and device drivers.
9	Provides reference information that is required to write device drivers in the kernel environment.

Man Page Format

Man pages use a standard format that includes some or all of the following subsections. The order of the subsections here matches the order in which the information appears in the individual man page sections.

`NAME` Provides the name of the command, library function, file, or device driver, including a brief description of what the construct does.

SYNOPSIS	Shows the syntax of the command, library function, file, or device driver.
DESCRIPTION	Defines the functionality and behavior of the service.
IOCTL	Used for a particular class of devices, all of which have an `io` ending, such as `mtio`(7I). This section appears on pages in Section 7 only.
OPTIONS	Lists the command options, including a concise summary of what each option does.
OPERANDS	Lists the command operands and describes how the operands affect the actions of the command.
OUTPUT	Describes the output, standard output, standard error, or output files that are generated by the command, library function, file, or device driver.
RETURN VALUES	Lists values and describes the conditions under which they are returned (applies only to man pages that document functions that return values).
ERRORS	Lists alphabetically all error codes that a function library can generate and describes the conditions that cause each error.
USAGE	Lists special rules, features, and commands that require in-depth explanations.
EXAMPLES	Provides examples of usage or how to use a command, library function, file, or device driver.
ENVIRONMENT VARIABLES	Lists any environment variables that the command, library function, file, or device driver affects, followed by a brief description of the effect.
EXIT STATUS	Lists the values that the command returns to the calling program or shell, and the conditions that cause these values to be returned.
FILES	Lists all file names that are referred to by the man page, including files of interest, and files that are created or required by commands.
ATTRIBUTES	Lists characteristics of commands, utilities, and device drivers by defining the attribute type and its corresponding value. See the `attributes(5)` man page.
SEE ALSO	Includes any related commands or library functions.
DIAGNOSTICS	Lists diagnostic messages with a brief explanation of the condition causing the error.

WARNINGS	Lists warnings about special conditions that could seriously affect working conditions. This is not a list of diagnostics.
NOTES	Lists additional information that does not belong anywhere else on the page.
BUGS	Describes known bugs and, wherever possible, suggests a workaround.

◆ ◆ ◆ CHAPTER 2

Managing User Accounts and Groups (Overview)

This is a list of the information that is in this chapter:

- "Removal of Support for Solaris Management Console" on page 33
- "What Are User Accounts and Groups?" on page 34
- "Where User Account and Group Information Is Stored" on page 41
- "Command-Line Tools for User and Group Account Management" on page 46
- "Customizing a User's Work Environment" on page 47

What's New in User Accounts and Groups?

The following features are new or changed in Oracle Solaris 11.

Removal of Support for Solaris Management Console

Note – The Solaris Management Console graphical tool that is used to manage users, groups, roles, and rights is no longer supported. All of the equivalent Solaris Management Console command-line tools are also not supported in Oracle Solaris 11. To create and manage users, groups, roles, and rights, use the command-line tools that are described or referenced within this chapter.

Default Password Hashing Algorithm

The default password hashing algorithm in Oracle Solaris 11 has been changed to SHA256. The password hash for the user is similar to the following:

```
$5$cgQk2iUy$AhHtVGx5Qd0.W3NCKjikb8.KhOiA4DpxsW55sP0UnYD
```

The eight character limitation on passwords applies only to passwords that use the older `crypts_unix`(5) algorithm, which has been preserved for backwards compatibility with any existing `passwd` file entries and NIS maps.

What Are User Accounts and Groups?

One basic system administration task is to set up a user account for each user at a site. A typical user account includes the information a user needs to log in and use a system, without having the system's `root` password. User account components are described in "User Account Components" on page 34.

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

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

A special type of user account, called a *role*, gives selected users special privileges. For more information, see "Role-Based Access Control (Overview)" in *Oracle Solaris Administration: Security Services*.

User Account Components

The following sections describe the various components of a user account.

User (Login) Names

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

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

If this scheme still results in duplicate names, consider using the following scheme to create a user name:

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

---

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

For detailed guidelines on setting up user (login) names, see "Guidelines for Assigning User Names, User IDs, and Group IDs" on page 40.

User ID Numbers

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

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

TABLE 2–1 Reserved UID Numbers

UID Numbers	User or Login Accounts	Description
0 – 99	`root`, `daemon`, `bin`, `sys`, and so on	Reserved for use by the operating system
100 – 2147483647	Regular users	General purpose accounts
60001 and 65534	`nobody` and `nobody4`	Anonymous users
60002	`noaccess`	Non trusted users

Do not assign UIDs 0 through 99. These UIDs are reserved for allocation by Oracle Solaris. By definition, `root` always has UID 0, `daemon` has UID 1, and pseudo-user `bin` has UID 2. In addition, you should give `uucp` logins and pseudo user logins, such as `who`, `tty`, and `ttytype`, low UIDs so that they fall at the beginning of the `passwd` file.

For additional guidelines on setting up UIDs, see "Guidelines for Assigning User Names, User IDs, and Group IDs" on page 40.

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

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

Using Large User IDs and Group IDs

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

The following table describes UID and GID limitations.

TABLE 2–2 Large UID and GID Limitation Summary

UID or GID	Limitations
262144 or greater	Users who use the `cpio` command with the default archive format to copy a file see an error message for each file. And, the UIDs and GIDs are set to `nobody` in the archive.
2097152 or greater	Users who use the `cpio` command with the `-H odc` format or the `pax -x cpio` command to copy files see an error message returned for each file. And, the UIDs and GIDs are set to `nobody` in the archive.
1000000 or greater	Users who use the `ar` command have their UIDs and GIDs set to `nobody` in the archive.
2097152 or greater	Users who use the `tar` command, the `cpio -H ustar` command, or the `pax -x tar` command have their UIDs and GIDs set to `nobody`.

UNIX Groups

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

Each group must have a name, a group identification (GID) number, and a list of user names that belong to the group. A GID number identifies the group internally to the system.

The two types of groups that a user can belong to are as follows:

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

For detailed guidelines on setting up group names, see "Guidelines for Assigning User Names, User IDs, and Group IDs" on page 40.

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

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

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

Groups can be local to a system or managed through a name service. To simplify group administration, you should use a name service such as NIS or a directory service such as LDAP. These services enable you to centrally manage group memberships.

User Passwords

You can specify a password for a user when you add the user. Or, you can force the user to specify a password when the user first logs in.

User passwords must comply with the following syntax:

- Password length must at least match the value identified by the `PASSLENGTH` variable in the `/etc/default/passwd` file. By default, `PASSLENGTH` is set to 6.
- The first 6 characters of the password must contain at least two alphabetic characters and have at least one numeric or special character.

Although user names are publicly known, passwords must be kept secret and known only to users. Each user account should be assigned a password.

Note – In Oracle Solaris 11, the default password hashing algorithm has been changed to SHA256. As a result, there is no longer an eight character limitation for user passwords as in previous Oracle Solaris releases. The eight character limitation only applies to passwords that use the older `crypts_unix`(5) algorithm, which has been preserved for backwards compatibility with any existing `passwd` file entries and NIS maps.

Passwords are now encoded by using one of the other `crypt`(3c) algorithms, including the SHA256 algorithm that is the default in the Solaris 11 `policy.conf` file. Thus, passwords can now be much longer than eight characters.

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

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

Good choices for passwords include the following:

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

Do not use these choices for passwords:

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

For task-related information, see "How to Add a User" on page 61.

Home Directories

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

A home directory can be located either on the user's local system or on a remote file server. In either case, by convention the home directory should be created as `/export/home/`*username*. For a large site, you should store home directories on a server. Use a separate file system for each user. For example, `/export/home/alice` or `/export/home/bob`. By creating separate file systems for each user, you can set properties or attributes based on each user's needs.

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

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

Naming Services

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

User's Work Environment

Besides having a home directory to create and store files, users need an environment that gives them access to the tools and resources they need to do their work. When a user logs in to a system, the user's work environment is determined by initialization files. These files are defined by the user's startup shell, which can vary, depending on the release.

A good strategy for managing the user's work environment is to provide customized user initialization files, such as `.bash_profile`, `.bash_login`, `.kshrc`, or `.profile`, in the user's home directory.

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

For detailed information about customizing user initialization files for users, see "Customizing a User's Work Environment" on page 47.

For information about how to customize user accounts through role-based access control (RBAC) feature of Oracle Solaris, see "Role-Based Access Control (Overview)" in *Oracle Solaris Administration: Security Services* for more information.

Guidelines for Assigning User Names, User IDs, and Group IDs

User names, UIDs, and GIDs should be unique within your organization, which could span multiple domains.

Keep the following guidelines in mind when creating user or role names, UIDs, and GIDs:

- **User names –** Should contain from two to eight letters and numerals. The first character should be a letter. At least one character should be a lowercase letter.

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

- **System accounts –** Do not use any of the user names, UIDs, or GIDs that are contained in the default `/etc/passwd` and `/etc/group` files. Do not use the UIDs and GIDs, 0-99. These numbers are reserved for allocation by Oracle Solaris and should not be used by anyone. Note that this restriction also applies to numbers not currently in use.

  For example, `gdm` is the reserved user name and group name for the GNOME Display Manager daemon and should not be used for another user. For a complete listing of the default `/etc/passwd` and `/etc/group` entries, see Table 2–3 and Table 2–4.

  The `nobody` and `nobody4` accounts should never be used for running processes. These two accounts are reserved for use by NFS. Use of these accounts for running processes could lead to unexpected security risks. Processes that need to run as a non-root user should use the `daemon` or `noaccess` accounts.

- **System account configuration** – The configuration of the default system accounts should never be changed. This includes changing the login shell of a system account that is currently locked. The only exception to this rule is the setting of a password and password aging parameters for the `root` account.

Note – Changing a password for a locked user account changes the password, but no longer unlocks the account at the same time. A second step to unlock the account by using the `passwd -u` command is now required.

Where User Account and Group Information Is Stored

Depending on your site policy, user account and group information can be stored in your local system's `/etc` files or in a name or directory service as follows:

- The NIS name service information is stored in maps.
- The LDAP directory service information is stored in indexed database files.

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

Most user account information is stored in the `passwd` file. Password information is stored as follows:

- In the `passwd` file when you are using NIS
- In the `/etc/shadow` file when you are using `/etc` files
- In the `people` container when you are using LDAP

Password aging is available when you are using LDAP, but not NIS.

Group information is stored in the `group` file for NIS, and files. For LDAP, group information is stored in the `group` container.

Fields in the `passwd` File

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

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

For example:

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

For a complete description of the fields in the `passwd` file, see the `passwd(1)` man page.

Default passwd File

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

```
root:x:0:0:Super-User:/root:/usr/bin/bash
daemon:x:1:1::/:
bin:x:2:2::/usr/bin:
sys:x:3:3::/:
adm:x:4:4:Admin:/var/adm:
lp:x:71:8:Line Printer Admin:/:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
dladm:x:15:65:Datalink Admin:/:
netadm:x:16:65:Network Admin:/:
netcfg:x:17:65:Network Configuration Admin:/:
smmsp:x:25:25:SendMail Message Submission Program:/:
listen:x:37:4:Network Admin:/usr/net/nls:
gdm:x:50:50:GDM Reserved UID:/var/lib/gdm:
zfssnap:x:51:12:ZFS Automatic Snapshots Reserved UID:/:/usr/bin/pfsh
upnp:x:52:52:UPnP Server Reserved UID:/var/coherence:/bin/ksh
xvm:x:60:60:xVM User:/:
mysql:x:70:70:MySQL Reserved UID:/:
openldap:x:75:75:OpenLDAP User:/:
webservd:x:80:80:WebServer Reserved UID:/:
postgres:x:90:90:PostgreSQL Reserved UID:/:/usr/bin/pfksh
svctag:x:95:12:Service Tag UID:/:
unknown:x:96:96:Unknown Remote UID:/:
nobody:x:60001:60001:NFS Anonymous Access User:/:
noaccess:x:60002:60002:No Access User:/:
nobody4:x:65534:65534:SunOS 4.x NFS Anonymous Access User:/:
ftp:x:21:21:FTPD Reserved UID:/:
dhcpserv:x:18:65:DHCP Configuration Admin:/:
aiuser:x:60003:60001:AI User:/:
pkg5srv:x:97:97:pkg(5) server UID:/:
```

TABLE 2–3 Default passwd File Entries

User Name	User ID	Description
root	0	Reserved for superuser account
daemon	1	Umbrella system daemon associated with routine system tasks
bin	2	Administrative daemon associated with running system binaries to perform some routine system task
sys	3	Administrative daemon associated with system logging or updating files in temporary directories
adm	4	Administrative daemon associated with system logging
lp	71	Reserved for the Line printer daemon

TABLE 2–3 Default passwd File Entries *(Continued)*

User Name	User ID	Description
uucp	5	Assigned to the daemon that is associated with uucp functions
nuucp	9	Assigned to another daemon associated with uucp functions
dladm	15	Reserved for datalink administration
netadm	16	Reserved for network administration
netcfg	17	Reserved for network configuration administration
smmsp	25	Assigned to the Sendmail message submission program daemon
listen	37	Assigned to the Network Listener daemon
gdm	50	Assigned to the GNOME Display Manager daemon
zfssnap	51	Reserved for automatic snapshots
upnp	52	Reserved for UPnP server
xvm	60	Reserved for xVM user
mysql	70	Reserved for MySQL user
openldap	75	Reserved for OpenLDAP user
webservd	80	Reserved for WebServer access
postgres	90	Reserved for PostgresSQL access
svctag	95	Reserved for Service Tag Registry access
unknown	96	Reserved for unmappable remote users in NFSv4 ACLs
nobody	60001	Reserved for NFS Anonymous Access user
noaccess	60002	Reserved for No Access user
nobody4	65534	Reserved for SunOS 4.x NFS Anonymous Access user
ftp	21	Reserved for FTP access
dhcpserv	18	Reserved for DHCP server user
aiuser	60003	Reserved for AI user
pkg5srv	97	Reserved for pkg(5) depot server

Fields in the shadow File

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

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

The default password hashing algorithm is SHA256. The password hash for the user is similar to the following:

```
$5$cgQk2iUy$AhHtVGx5Qd0.W3NCKjikb8.KhOiA4DpxsW55sP0UnYD
```

For a complete description of the fields in the shadow file, see the shadow(4) man page.

Fields in the group File

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

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

For example:

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

For a complete description of the fields in the group file, see the group(4) man page.

Default group File

The default group file contains the following system groups that support some system-wide task, such as printing, network administration, or electronic mail. Most of these groups have corresponding entries in the passwd file.

```
root::0:
other::1:root
bin::2:root,daemon
sys::3:root,bin,adm
adm::4:root,daemon
uucp::5:root
mail::6:root
tty::7:root,adm
lp::8:root,adm
nuucp::9:root
staff::10:
daemon::12:root
sysadmin::14:
games::20:
smmsp::25:
gdm::50:
upnp::52:
```

```
xvm::60:
netadm::65:
mysql::70:
openldap::75:
webservd::80:
postgres::90:
slocate::95:
unknown::96:
nobody::60001:
noaccess::60002:
nogroup::65534:
ftp::21
pkg5srv::97:
```

TABLE 2–4 Default `group` File Entries

Group Name	Group ID	Description
`root`	`0`	Superuser group
`other`	`1`	Optional group
`bin`	`2`	Administrative group associated with running system binaries
`sys`	`3`	Administrative group associated with system logging or temporary directories
`adm`	`4`	Administrative group associated with system logging
`uucp`	`5`	Group associated with `uucp` functions
`mail`	`6`	Electronic mail group
`tty`	`7`	Group associated with `tty` devices
`lp`	`8`	Line printer group
`nuucp`	`9`	Group associated with `uucp` functions
`staff`	`10`	General administrative group.
`daemon`	`12`	Group associated with routine system tasks
`sysadmin`	`14`	Administrative group that is useful for system administrators
`smmsp`	25	Daemon for Sendmail message submission program
`gdm`	50	Group reserved for the GNOME Display Manager daemon
`upnp`	`52`	Group reserved for UPnP server
`xvm`	`60`	Group reserved for xVM user
`netadm`	`65`	Group reserved for network administration
`mysql`	`70`	Group reserved for MySQL user

TABLE 2–4 Default `group` File Entries *(Continued)*

Group Name	Group ID	Description
`openldap`	`75`	Reserved for OpenLDAP user
`webservd`	`80`	Group reserved for WebServer access
`postgres`	`90`	Group reserved for PostgresSQL access
`slocate`	`95`	Group reserved for Secure Locate access
`unknown`	`96`	Group reserved for unmappable remote groups in NFSv4 ACLs
`nobody`	`60001`	Group assigned for anonymous NFS access
`noaccess`	`60002`	Group assigned to a user or a process that needs access to a system through some application but without actually logging in
`nogroup`	`65534`	Group assigned to a user who is not a member of a known group
`ftp`	`21`	Group assigned for FTP access
`pkg5srv`	`97`	Group assigned to `pkg`(5) depot server

Command-Line Tools for User and Group Account Management

Note – The Solaris Management Console and all its equivalent command-line tools are no longer supported.

The following commands are available for managing user and group accounts.

TABLE 2–5 Commands to Manage Users and Groups

Man Page for Command	Description	For Additional Information
`useradd`(1M)	Creates users locally or in an LDAP repository.	"How to Add a User" on page 61
`usermod`(1M)	Changes user properties locally or in an LDAP repository. If the user properties are security-relevant, such as role assignment, this task might be restricted to your security administrator or to the `root` role.	"How to Change the RBAC Properties of a User" in *Oracle Solaris Administration: Security Services*
`userdel`(1M)	Deletes a user from the system or from the LDAP repository. Can involve additional cleanup, such as `cron` job removal.	"How to Delete a User" on page 62

TABLE 2–5 Commands to Manage Users and Groups *(Continued)*

Man Page for Command	Description	For Additional Information

`groupadd(1M)`
`groupmod(1M)`
`groupdel(1M)`
	Manages groups locally or in an LDAP repository.	"How to Add a Group" on page 63

`roleadd(1M)`
`rolemod(1M)`
`roledel(1M)`
	Manages roles locally or in an LDAP repository. Roles cannot log in. Users assume an assigned role to perform administrative tasks.	"How to Create a Role" in *Oracle Solaris Administration: Security Services*
		"Initially Configuring RBAC (Task Map)" in *Oracle Solaris Administration: Security Services*

Customizing a User's Work Environment

Part of setting up a user's home directory is providing user initialization files for the user's login shell. A *user initialization file* is a shell script that sets up a work environment for a user after the user logs in to a system. Basically, you can perform any task in a user initialization file that you can do in a shell script. However, a user initialization file's primary job is to define the characteristics of a user's work environment, such as a user's search path, environment variables, and windowing environment. Each login shell has its own user initialization file, or files, which are listed in the following table. Note that the default user initialization file for both the bash and `ksh93` shells is `/etc/skel/local.profile`.

TABLE 2–6 Bash and `ksh93` User Initialization Files

Shell	User Initialization File	Purpose
bash	`$HOME/.bash_profile`	Defines the user's environment at login
	`$HOME/.bash_login`	
	`$HOME/.profile`	
ksh93	`/etc/profile`	Defines the user's environment at login
	`$HOME/.profile`	
	`$ENV`	Defines user's environment at login in the file and is specified by the Korn shell's `ENV` environment variable

You can use these files as a starting point and then modify them to create a standard set of files that provide the work environment common to all users. You can also modify these files to provide the working environment for different types of users.

For step-by-step instructions on how to create sets of user initialization files for different types of users, see "How to Customize User Initialization Files" on page 60.

Using Site Initialization Files

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

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

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

To reference a site initialization file in a bash or `ksh93` user initialization file, place a line at the beginning of the user initialization file similar to the following line:

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

Avoiding Local System References

Do not add specific references to the local system in the user initialization file. The instructions in a user initialization file should be valid, regardless of which system the user logs into.

For example:

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

Shell Features

The user account that is created when you install the Oracle Solaris release is assigned the GNU Bourne-Again Shell (bash) by default. The standard system shell, `bin/sh`, is now the Korn Shell 93 (`ksh93`). The default interactive shell is shell is the Bourne-again (`bash`) shell, `/usr/bin/bash`. Both the bash and `ksh93` shells feature command-line editing, which means you can edit commands before executing them. To change to a different shell, type the path of the shell that you want to use. To exit a shell, type `exit`.

The following table describes the shell options that are supported in this release.

TABLE 2–7 Basic Shell Features in the Oracle Solaris Release

Shell	Path	Comments
Bourne-Again Shell (bash)	`/usr/bin/bash`	Default shell for users that are created by an installer, as well as the `root` role The default (interactive) shell for users that are created with the `useradd` command, as well as the `root` role, is `/usr/bin/bash`. The default path is `/usr/bin:/usr/sbin`.
Korn Shell	`/usr/bin/ksh`	`ksh93` is the default shell in this Oracle Solaris release
C Shell and enhanced C Shell	`/usr/bin/csh` and `/usr/bin/tcsh`	C Shell and enhanced C Shell
POSIX-compliant Shell	`/usr/xpg4/bin/sh`	POSIX-compliant shell
Z Shell	`/usr/bin/zsh`	Z Shell

Note – The Z Shell (`zsh`) and the enhanced C Shell (`tsch`) are not installed on your system by default. To use either of these shells, you must first install the required software packages.

Bash and `ksh93` Shell History

Both the `bash` and `ksh93` shells record a history of all of the commands that you run. This history is kept on a per user basis, which means history is persistent between login sessions and is representative of all your login sessions.

For example, if you are in a bash shell, to see the complete history of commands that you have run, you would type:

```
$ history
1 ls
2 ls -a
3 pwd
4 whoami
.
.
.
```

To display a number of previous commands, include an integer in the command:

```
$ history 2
12 date
13 history
```

For more information, see the `history(1)` man page.

Bash and `ksh93` Shell Environment Variables

The bash and `ksh93` shells store special variable information that is known to the shell as an*environment variable*. To view a complete list of the current environment variables for the bash shell, use the `declare` command as follows:

```
$ declare
BASH=/usr/bin/bash
BASH_ARGC=()
BASH_ARGV=()
BASH_LINEND=()
BASH_SOuRCE=()
BASH_VERSINFO=([0]=''3'' [1]=''2'' [2]=''25'' [3]=''1''
[4]=''release'' [5]'')
.
.
.
```

For the `ksh93` shell, use the `set` command, which is the bash shell's `declare` command equivalent:

```
$ set
  COLUMNS=80
  ENV='$HOME/.kshrc'
  FCEDIT=/bin/ed
  HISTCMD=3
  HZ=''
  IFS=$' \t\n'
  KSH_VERSION=.sh.version
  LANG=C
  LINENO=1
  .
  .
  .
```

To print environment variables for either shell, use the `echo` or `printf` command. For example:

```
$ echo $SHELL
/usr/bin/bash
$ printf ''$PATH\n''
/usr/bin:/usr/sbin
```

Note – Environment variables do not persist between sessions. To set up environment variables that remain consistent between logins, you must make the changes in the `.bashrc` file.

A shell can have two types of variables:

Environment variables
: Specifies variables that are exported to all processes that are spawned by the shell. The `export` command is used to export a variable. For example:

  ```
  export VARIABLE=value
  ```

  These settings can be displayed by using the `env` command. A subset of environment variables, such as `PATH`, affects the behavior of the shell itself.

Shell (local) variables
: Specifies variables that affect only the current shell.

  In a user initialization file, you can customize a user's shell environment by changing the values of the predefined variables or by specifying additional variables.

The following table provides more details about the shell and environment variables that are available in the Oracle Solaris release.

TABLE 2–8 Shell and Environment Variable Descriptions

Variable	Description
`CDPATH`	Sets a variable used by the `cd` command. If the target directory of the `cd` command is specified as a relative path name, the `cd` command first looks for the target directory in the current directory (`.`). If the target is not found, the path names listed in the `CDPATH` variable are searched consecutively until the target directory is found and the directory change is completed. If the target directory is not found, the current working directory is left unmodified. For example, the `CDPATH` variable is set to `/home/jean`, and two directories exist under `/home/jean`, `bin`, and `rje`. If you are in the `/home/jean/bin` directory and type `cd rje`, you change directories to `/home/jean/rje`, even though you do not specify a full path.
`HOME`	Sets the path to the user's home directory.
`LANG`	Sets the locale.

TABLE 2–8 Shell and Environment Variable Descriptions *(Continued)*

Variable	Description
LOGNAME	Defines the name of the user currently logged in. The default value of LOGNAME is set automatically by the login program to the user name specified in the passwd file. You should only need to refer to, not reset, this variable.
MAIL	Sets the path to the user's mailbox.
MANPATH	Sets the hierarchies of man pages that are available.
PATH	Specifies, in order, the directories that the shell searches to find the program to run when the user types a command. If the directory is not in the search path, users must type the complete path name of a command. As part of the login process, the default PATH is automatically defined and set as specified in .profile. The order of the search path is important. When identical commands exist in different locations, the first command found with that name is used. For example, suppose that PATH is defined in the shell syntax as PATH=/usr/bin:/usr/sbin:$HOME/bin and a file named sample resides in both /usr/bin and /home/jean/bin. If the user types the command sample without specifying its full path name, the version found in /usr/bin is used.
PS1	Defines the shell prompt for the bash or ksh93 shell.
SHELL	Sets the default shell used by make, vi, and other tools.
TERMINFO	Names a directory where an alternate terminfo database is stored. Use the TERMINFO variable in either the /etc/profile or /etc/.login file. For more information, see the terminfo(4)man page. When the TERMINFO environment variable is set, the system first checks the TERMINFO path defined by the user. If the system does not find a definition for a terminal in the TERMINFO directory defined by the user, it searches the default directory, /usr/share/lib/terminfo, for a definition. If the system does not find a definition in either location, the terminal is identified as "dumb."
TERM	Defines the terminal. This variable should be reset in either the /etc/profile or /etc/.login file. When the user invokes an editor, the system looks for a file with the same name that is defined in this environment variable. The system searches the directory referenced by TERMINFO to determine the terminal characteristics.
TZ	Sets the time zone. The time zone is used to display dates, for example, in the ls -l command. If TZ is not set in the user's environment, the system setting is used. Otherwise, Greenwich Mean Time is used.

Customizing the Bash Shell

To customize your bash shell, add the information to the `.bashrc` file that is located in your home directory. The initial user that is created when you install Oracle Solaris has a `.bashrc` file that sets the `PATH`, `MANPATH`, and command prompt. For more information, see the `bash`(1) man page.

About the `MANPATH` Environment Variable

The `MANPATH` environment variable specifies where the `man` command looks for reference manual (man) pages. The `MANPATH` is set automatically based on a user's `PATH` value, but it generally includes `/usr/share/man` and `usr/gnu/share/man`.

Note that a user's `MANPATH` environment variable can be modified independent of the `PATH` environment variable. A one to one equivalent of the associated man page locations, with directories in the user's `$PATH`, is not required.

The `PATH` Variable

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

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

Setting Path Guidelines

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

- If you must include the current directory (`.`) in your path, it should be placed last. Including the current directory in your path is a security risk because some malicious person could hide a compromised script or executable in the current directory. Consider using absolute path names instead.
- Keep the search path as short as possible. The shell searches each directory in the path. If a command is not found, long searches can slow down system performance.
- The search path is read from left to right, so you should put directories for commonly used commands at the beginning of the path.
- Make sure that directories are not duplicated in the path.
- Avoid searching large directories, if possible. Put large directories at the end of the path.

- Put local directories before NFS mounted directories to lessen the chance of "hanging" when the NFS server does not respond. This strategy also reduces unnecessary network traffic.

Locale Variables

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

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

Note – By default, Oracle Solaris 11 installs UTF-8 based locales only.

The following table describes the environment variable values for the core Oracle Solaris 11 locales.

TABLE 2–9 Values for `LANG` and `LC` Variables

Value	Locale
`en_US.UTF-8`	English, United States (UTF-8)
`fr_FR.UTF-8`	French, France (UTF-8)
`de_DE.UTF-8`	German, Germany (UTF-8)
`it_IT.UTF-8`	Italian, Italy (UTF-8)
`ja_JP.UTF-8`	Japanese, Japan (UTF-8)
`ko_KR.UTF-8`	Korean, Korea (UTF-8)
`pt_BT.UTF-8`	Portuguese, Brazil (UTF-8)
`zh_CN.UTF-8`	Simplified Chinese, China (UTF–8)
`es_ES.UTF-8`	Spanish, Spain (UTF-8)
`zh_TW.UTF-8`	Traditional Chinese, Taiwan (UTF-8)

EXAMPLE 2–1 Setting the Locale Using the `LANG` Variables

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

```
LANG=de_DE.ISO8859-1; export LANG
```

LANG-de_DE.UTF-8; export LANG

Default File Permissions (`umask`)

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

The user mask contains the following octal values:

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

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

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

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

TABLE 2–10 Permissions for `umask` Values

`umask` Octal Value	File Permissions	Directory Permissions
`0`	`rw-`	`rwx`
1	`rw-`	`rw-`
2	`r--`	`r-x`
3	`r--`	`r--`
4	`-w-`	`-wx`
5	`-w-`	`-w-`

TABLE 2–10 Permissions for `umask` Values *(Continued)*

umask Octal Value	File Permissions	Directory Permissions
6	`--x`	`--x`
7	`---` (none)	`---` (none)

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

```
umask 000
```

Customizing a User Initialization File

The following is an example of the `.profile` user initialization file. You can use this file to customize your own user initialization files. This example uses system names and paths that you will need to modify for your particular site.

EXAMPLE 2–2 The `.profile` File

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

1. Defines the user's shell search path.
2. Defines the path to the user's mail file.
3. Defines the user's time/clock server.
4. Defines the user's search path for man pages.
5. Defines the user's default printer.
6. Sets the user's default file creation permissions.
7. Sets the listed environment variables.

◆ ◆ ◆ CHAPTER 3

Managing User Accounts and Groups (Tasks)

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

For background information about managing user accounts and groups, see Chapter 2, "Managing User Accounts and Groups (Overview)."

Setting Up and Administering User Accounts (Task Map)

Task	Description	For Instructions
Gather user information.	Use a standard form to gather user information to help you keep user information organized.	"Gathering User Information" on page 59
Customize user initialization files.	You can set up user initialization files, so that you can provide new users with consistent environments.	"How to Customize User Initialization Files" on page 60
Change account defaults for all roles.	Change the default home directory and skeleton directory for all roles.	"How to Change Account Defaults For All Roles" on page 60
Create a user account.	Using the account defaults that you set up, create a local user by using the `useradd` command.	"How to Add a User" on page 61
Delete a user account.	You can delete a user account by using the `userdel` command.	"How to Delete a User" on page 62

Task	Description	For Instructions
Create, then assign a role to perform an administrative task.	Using the account defaults that you set up, create a local role, so that the user can perform a specific administrative command or task.	"How to Create a Role" in *Oracle Solaris Administration: Security Services* "How to Assign a Role" in *Oracle Solaris Administration: Security Services*
Create a group.	To create a new group, use the `groupadd` command.	"How to Add a Group" on page 63
Add security attributes to a user account.	After you set up a local user account, you can add the required security attributes.	"How to Change the RBAC Properties of a User" in *Oracle Solaris Administration: Security Services*
Share a user's home directory.	You must share the user's home directory, so that the directory can be remotely mounted from the user's system.	"How to Share Home Directories That Are Created as ZFS File Systems" on page 63
Manually mount a user's home directory.	Typically, you do not need to manually mount user home directories that are created as a ZFS file system. The home directory is mounted automatically when it is created and also at boot time from the SMF local file system service.	"Manually Mounting a User's Home Directory." on page 64

Setting Up User Accounts

In Oracle Solaris 11, user accounts are created as Oracle Solaris ZFS file systems. As an administrator, when you create user accounts, you are creating more than a home directory. You are giving users their own file system and their own ZFS dataset. Every home directory that is created by using the `useradd` and `roleadd` commands places the home directory of the user on the `/export/home` file system as an *individual* ZFS file system. As a result, users have the ability to back up their home directories, create ZFS snapshots of their home directories, and replace files in their current home directory from the ZFS snapshots that they created.

The `useradd` command relies on the automount service, `svc:/system/filesystem/autofs` to mount a user's home directory, so this service should not be disabled. Each home directory entry for a user in the `passwd` database is of the form `/home/`*username*, which is an `autofs` trigger that is resolved by the automounter through the `auto_home` map.

The `useradd` command automatically creates entries in the `auto_home` map that correspond to the pathname that is specified by using the `-d` option. If the pathname includes a remote host specification, for example, `foobar:/export/home/jdoe`, then the home directory for `jdoe` must be created on the system `foobar`. The default pathname is `localhost:/export/home/`*user*.

Because this file system is a ZFS dataset, the user's home directory is created as a child ZFS dataset, with the ZFS permission to take snapshots delegated to the user. If a pathname is specified that does not correspond to a ZFS dataset, then a regular directory is created. If the `-S ldap` option is specified, then the `auto_home` map entry is updated on the LDAP server instead of the local `auto_home` map.

Gathering User Information

When setting up user accounts you can create a form similar to the following form to gather information about users before adding their accounts.

Item	Description
User Name:	
Role Name:	
Profiles or Authorizations:	
UID:	
Primary Group:	
Secondary Groups:	
Comment:	
Default Shell:	
Password Status and Aging:	
Home Directory Path Name:	
Mounting Method:	
Permissions on Home Directory:	
Mail Server:	
Department Name:	
Department Administrator:	
Manager:	
Employee Name:	
Employee Title:	
Employee Status:	
Employee Number:	

Item	Description
Start Date:	
Add to These Mail Aliases:	
Desktop System Name:	

How to Customize User Initialization Files

1. **Become the `root` role.**

```
$ su -
Password:
#
```

Note – This method works whether `root` is a user account or a role.

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

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

shared-dir The name of a directory that is available to other systems on the network.

user-type The name of a directory to store initialization files for a type of user.

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

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

For a detailed description on the ways to customize the user initialization files, see "Customizing a User's Work Environment" on page 47.

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

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

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

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

How to Change Account Defaults For All Roles

In the following procedure, the administrator has customized a `roles` directory . The administrator changes the default home directory and skeleton directory for all roles.

1. **Become the `root` role.**

2. **Create a custom roles directory. For example:**

```
# roleadd -D
group=other,1  project=default,3  basedir=/home
skel=/etc/skel  shell=/bin/pfsh  inactive=0
expire=  auths=  profiles=All  limitpriv=
defaultpriv=  lock_after_retries=
```

3. **Change the default home directory and skeleton directory for all roles. For example:**

```
# roleadd -D -b /export/home -k /etc/skel/roles
# roleadd -D
group=staff,10  project=default,3  basedir=/export/home
skel=/etc/skel/roles  shell=/bin/sh  inactive=0
expire=  auths=  profiles=  roles=  limitpriv=
defaultpriv=  lock_after_retries=
```

Future uses of the **`roleadd`** command create home directories in `/export/home`, and populate the roles' environment from the `/etc/skel/roles` directory.

How to Add a User

In Oracle Solaris 11, user accounts are created as Oracle Solaris ZFS file systems. Every home directory that is created by using the `useradd` and `roleadd` commands places the home directory of the user on the `/export/home` file system as an *individual* ZFS file system.

1. **Become the `root` role.**

2. **Create a local user.**

By default, the user is created locally. With the `-S ldap` option, the user is created in an existing LDAP repository.

```
# useradd -m username
```

`useradd`	Creates an account for the specified user.
`-m`	Creates a local home directory on the system for the specified user.

Note – The account is locked until you assign the user a password.

3. **Assign the user a password.**

```
$ passwd username
New password:       Type user password
Re-enter new password:       Retype password
```

For more command options, see the `useradd(1M)` and `passwd(1)` man pages.

See Also After creating a user, if you want to add roles or assign roles to the user account, see "How to Create a Role" in *Oracle Solaris Administration: Security Services* for more information.

How to Delete a User

1. **Become the `root` role.**

```
$ su -
Password:
#
```

Note – This method works whether `root` is a user account or a role.

2. **Archive the user's home directory.**

3. **Run one of the following commands:**

   - **If the user has a local home directory, delete the user and the home directory.**

     `# userdel -r` *username*

     `usesrdel` Deletes the account of the specified user.

     `-r` Removes the account from the system.

     Because user home directories are now ZFS datasets, the preferred method for removing a local home directory for a deleted user is to specify the `-r` option with the `userdel` command.

   - **Otherwise, delete the user only.**

     `# userdel` *username*

     You must manually delete the user's home directory on the remote server.

   For a full list of command options, see the `userdel(1M)` man page.

Next Steps Additional cleanup might be required if the user that you deleted had administrative responsibilities, for example creating `cron` jobs, or if the user had additional accounts in non-global zones.

How to Add a Group

1. **Become the `root` role.**

2. **List the existing groups.**

   ```
   # cat /etc/group
   ```

3. **Create a new group.**

   ```
   $ groupadd -g 18 exadata
   ```

   `groupadd` Creates a new group definition on the system by adding the appropriate entry to the `/etc/group` file.

   `-g` Assigns the group ID for the new group.

   For more information, see the `groupadd(1M)` man page.

Example 3–1 Adding a Group and User With the `groupadd` and `useradd` Commands

The following example shows how to use the `groupadd` and `useradd` commands to add the group `scutters` and the user `scutter1` to files on the local system.

```
# groupadd -g 102 scutters
# useradd -u 1003 -g 102 -d /export/home/scutter1 -s /bin/csh \
-c "Scutter 1" -m -k /etc/skel scutter1
64 blocks
```

For more information, see the `groupadd(1M)` and `useradd(1M)` man pages.

How to Share Home Directories That Are Created as ZFS File Systems

An NFS or a Server Message Block (SMB) share of a ZFS file system is created and then the share is published.

The two-step process is as follows:

- The file system share is created by using `zfs set share` command. At this time, specific share properties can be defined. If share properties are not defined, the default property values are used for the share.
- The NFS or SMB share is published by setting the `sharenfs` or `sharesmb` property. The share is published permanently until the property is set to `off`.

Note that you must be the root user to perform the following procedure.

Before You Begin Create a separate pool for the user home directories. For example:

```
# zpool create users mirror c1t1d0 c1t2d0 mirror c2t1d0 c2t2d0
```

Then, create a container for the home directories:

```
# zfs create /users/home
```

1. **Become the `root` role.**

2. **Create the share and set the `sharenfs` or `sharesmb` properties. For example, to create an NFS share and set the `sharenfs` property, type the following commands:**

```
# zfs set share=name=users,path=/users/home,prot=nfs users/home
name=users,path=/users/home,prot=nfs
# zfs set sharenfs=on users/home
```

3. **Create the individual file systems that will inherit the share property options and the `sharenfs` or `sharesmb` property.**

```
# zfs create users/home/username1
# zfs create users/home/username2
# zfs create users/home/username3
```

For example:

```
# zfs create users/home/alice
# zfs create users/home/bob
# zfs create users/home/carl
```

4. **Determine whether the ZFS file system shares are published.**

```
# zfs get -r sharenfs users/home
```

The `-r` option displays all of the descendent file systems.

See Also For more information about creating and publishing shares, see "Mounting ZFS File Systems" in *Oracle Solaris Administration: ZFS File Systems*.

Manually Mounting a User's Home Directory.

User accounts that are created as ZFS file systems do not typically need to be manually mounted. With ZFS, file systems are automounted when they are created and then mounted at boot time from the SMF local file system service.

When creating user accounts, make sure home directories are set up as they are in the name service, at `/home/`*username*. Then, make sure that the `auto_home` map indicates the NFS path to the user's home directory. For task-related information, see "Task Overview for Autofs Administration" in *Oracle Solaris Administration: Network Services*.

If you need to manually mount a user's home directory, use the `zfs mount` command. For example:

```
# zfs mount users/home/alice
```

Note – Make sure that the user's home directory is shared. For more information, see "How to Share Home Directories That Are Created as ZFS File Systems" on page 63.

◆ ◆ ◆ CHAPTER 4

Booting and Shutting Down an Oracle Solaris System

Oracle Solaris is designed to run continuously so that electronic mail and network resources are available to users. This chapter provides a general overview and basic tasks for booting and shutting down SPARC and x86 based systems.

This is a list of the information that in this chapter:

- "What's New in Booting and Shutting Down a System?" on page 67
- "Booting and Shutting Down an Oracle Solaris System (Overview)" on page 70
- "Booting a System to a Specified State (Task Map)" on page 72
- "Shutting Down a System (Task Map)" on page 76
- "Booting a System From the Network" on page 79
- "Accelerating the Reboot Process (Task Map)" on page 80
- "Booting From a ZFS Boot Environment (Task Map)" on page 83
- "Modifying Boot Parameters (Task Map)" on page 87
- "Keeping a System Bootable (Task Map)" on page 93
- "Where to Find More Information About Booting and Shutting Down a System" on page 96

For detailed information about booting a SPARC based system, see *Booting and Shutting Down Oracle Solaris on SPARC Platforms*.

For detailed information about booting an x86 based system, see *Booting and Shutting Down Oracle Solaris on x86 Platforms*.

What's New in Booting and Shutting Down a System?

The following features are new Oracle Solaris 11:

- "Support for Administratively Provided `driver.conf` Files" on page 68
- "Bitmapped Console" on page 69
- "Boot and Shutdown Progress Animation" on page 69
- "x86: Removal of Support for 32–Bit Kernel" on page 70

Support for Administratively Provided `driver.conf` Files

In this Oracle Solaris release, vendor-provided `driver.conf` files can be supplemented with administratively provided `driver.conf` files. The format of an administratively provided `driver.conf` file is identical to a vendor-provided `driver.conf` file. Vendor-provided driver data is installed in the root file system, while administratively provided driver data is stored separately in a new `/etc/driver/drv` directory.

At boot time, and whenever a `driver.conf` file for a driver is searched for and loaded, the system checks for a configuration file in the `/etc/driver/drv` directory for that driver. If found, the system automatically merges the vendor-provider `driver.conf` files with the local, administratively provided `driver.conf` files. Note that the driver's view of the system properties consists of these merged properties. Therefore, no driver changes are necessary.

To display the merged properties, use the `prtconf` command with the new `-u` option. The `-u` option enables you to display both the original and updated property values for a specified driver. For more information, see the `prtconf(1M)` man page and "How to Display Default and Customized Property Values for a Device" on page 157.

Note – Do not edit vendor-provided `driver.conf` files that are located in the `/kernel` and `/platform` directories. If you need to supplement a driver's configuration, the preferred method is to add a corresponding `driver.conf` file to the local `/etc/driver/drv` directory, and then customize that file.

One advantage of customizing the administratively provided configuration file rather than the vendor-provide configuration file is that your changes are preserved during a system upgrade. During a system upgrade, if a vendor-provided `driver.conf` file has an update available, the file is automatically updated, and all customization is lost. Since there is no way to know which driver configuration files will be updated prior to performing an upgrade, always make it a practice to make any customization to the administratively provided version of the file. Before customizing an administratively provided configuration file, familiarize yourself with the `driver.conf` file format. See the `driver.conf(4)` man page for more information.

For detailed instructions, see Chapter 5, "Managing Devices (Overview/Tasks)," in *Oracle Solaris Administration: Devices and File Systems*.

Device driver writers should note that driver interfaces are provided to enable a driver to access both the `vendor` and `admin` properties. For more information, see the `driver(4)` man page and *Writing Device Drivers*.

For instructions, see the `ddi_prop_exists(9F)` and `ddi_prop_lookup(9F)` man pages.

Bitmapped Console

Oracle Solaris 11 supports higher resolution and color depth on x86 based systems than the older Video Graphics Array (VGA) 640-480 16-color console. This support is provided for systems that use traditional BIOS and Video Electronics Standards Association (VESA) option read-only memory (ROM). Note that support is limited to when a graphics card or frame buffer is used as a physical or virtual console. There is no impact on the behavior of serial consoles.

For more information, see "Support for Bitmapped Console" in *Booting and Shutting Down Oracle Solaris on x86 Platforms*.

Boot and Shutdown Progress Animation

The progress status indicator that is displayed on a system during the boot process is automatically interrupted in the following instances:

- Kernel debugger is entered
- System panic occurs
- An SMF service that requires input interrupts the boot process
- GNOME Desktop Manager (GDM) login screen displays

During the shutdown process, if the `console=graphics` option was specified when booting the system, and the shutdown is triggered by the `Xorg` server, a progress status indicator is displayed. You can prevent the progress status indicator from displaying by setting the new `splash-shutdown` property of the `svc:/system/boot-config` SMF service to `false`.

Fast Reboot

The Fast Reboot feature is supported on both the SPARC and x86 platform. The integration of Fast Reboot on the SPARC platform enables the `-f` option to be used with the `reboot` command to accelerate the boot process by skipping certain POST tests. On the x86 platform, Fast Reboot implements an in-kernel boot loader that loads the kernel into memory and then switches to that kernel. The firmware and boot loader processes are bypassed, which enables the system to reboot within seconds.

On both the x86 and SPARC platforms, the Fast Reboot feature is managed by SMF and implemented through a boot configuration service, `svc:/system/boot-config`. The `boot-config` service provides a means for setting or changing the default boot configuration parameters. When the `config/fastreboot_default` property is set to `true`, the system performs a fast reboot automatically, without the need to use the `reboot -f` command. This property's value is set to `false` on the SPARC platform and `true` on the x86 platform. For task-related information, including how to change the default behavior of Fast Reboot on the SPARC platform, see "Accelerating the Reboot Process" on page 81.

x86: Removal of Support for 32–Bit Kernel

In Oracle Solaris 11, 32–bit kernel support on x86 platforms has been removed. As a result, you cannot boot Oracle Solaris 11 on 32-bit x86 hardware. Systems that have 32-bit hardware must either be upgraded to 64–bit hardware or continue to run Oracle Solaris 10.

Note – This removal of support does not impact 32–bit applications, which remains the same as in previous releases.

Booting and Shutting Down an Oracle Solaris System (Overview)

The Oracle Solaris x86 and SPARC boot architectures share the following fundamental characteristics:

- **Use of a boot archive**

  The boot archive is a ramdisk image that contains all of the files that are required for booting a system. For more information, see "Description of the Oracle Solaris Boot Archives" in *Booting and Shutting Down Oracle Solaris on SPARC Platforms*.

- **Use of a boot administration interface to maintain the integrity of the Oracle Solaris boot archives**

  The `bootadm` command handles the details of boot archive update and verification. During an installation or upgrade, the `bootadm` command creates an initial boot archive. During the process of a normal system shutdown, the shutdown process compares the boot archive's contents with the root file system. If there have been updates to the system such as drivers or configuration files, the boot archive is rebuilt to include these changes so that upon reboot, the boot archive and root file system are synchronized. You can use the `bootadm` command to manually update the boot archive. For instructions, see "Maintaining the Integrity of the Boot Archives" in *Booting and Shutting Down Oracle Solaris on SPARC Platforms*.

  For more information, see the `bootadm`(1M) and `boot`(1M) man pages.

- **Use of a ramdisk image as the root file system during installation**

  The ramdisk image is derived from the boot archive and then transferred to the system from the boot device.

  In the case of a software installation, the ramdisk image is the root file system that is used for the entire installation process. The ramdisk file system type can be a High Sierra File System (HSFS).

For more information about SPARC boot processes, see "Description of the SPARC Boot Process" in *Booting and Shutting Down Oracle Solaris on SPARC Platforms*.

For more information about boot processes on the x86 platform, see "How the x86 Boot Process Works" in *Booting and Shutting Down Oracle Solaris on x86 Platforms*.

GRUB Based Booting

In Oracle Solaris, the open source GRand Unified Bootloader (GRUB) is the default boot loader on the x86 platform. GRUB is responsible for loading a boot archive into the system's memory. A boot archive is a collection of critical files that is needed during system startup before the root file system is mounted. The boot archive is the interface that is used to boot Oracle Solaris.

GRUB implements a menu interface that includes boot options that are predefined in a configuration file called the `menu.lst` file. GRUB also has a command-line interface that is accessible from the GUI menu interface that can be used to perform various boot functions, including modifying default boot parameters.

The menu that is displayed when you boot an x86 based system is the *GRUB menu*. This menu is based on configuration information that is in the GRUB `menu.lst` file. When the boot sequence starts, the GRUB menu is displayed. Unless you interrupt the boot sequence, the default entry (typically the first entry in the `menu.lst` file) is booted by default.

You can edit the GRUB menu at boot time to either boot a different operating system or modify the parameters of the default boot entry. To do so, type `e` as soon as the GRUB menu is displayed. Typing `e` interrupts the boot process and takes you to the *GRUB edit menu*, where you can select another OS to boot or modify default boot parameters for the default boot entry. Note that the modified boot behavior persists only until the next time the system is booted.

For task-related information, see *Booting and Shutting Down Oracle Solaris on x86 Platforms*.

Management of Boot Services by the Service Management Facility

With the introduction of SMF, the boot process now creates fewer messages. Also, services do not display a message by default when they are started. All of the information that was provided by the boot messages can now be found in a log file for each service that is in `/var/svc/log`. You can use the `svcs` command to help diagnose boot problems. To generate a message when each service is started during the boot process, use the `-v` option with the `boot` command.

Most of the features that are provided by SMF occur behind the scenes, so users are not typically aware of these features. Other features are accessed by new commands.

For more information, see "SMF and Booting" on page 114.

Booting a System to a Specified State (Task Map)

TABLE 4–1 Booting a System to a Specified State: Task Map

Task	Description	For Instructions
Determine the current run level of a system.	Use the `who` command with the `-r` option to determine a system's current run level.	"Determining a System's Current Run Level" on page 73
Boot a SPARC based system to a multiuser state.	Use this boot method to bring the system back to a multiuser state (run level 3) after shutting down or performing a system hardware maintenance task.	"SPARC: How to Boot a System to a Multiuser State (Run Level 3)" on page 73
Boot an x86 based system to a single-user state.	Use this boot method to perform a system maintenance task, such as backing up a file system.	"x86: How to Boot a System to a Single-User State (Run Level S)" on page 74

Booting a System to a Specified State (Run Level)

A system's *run level* (also known as an *init state*) defines what services and resources are available to users when the system is in that state. A system can be in only one run level at a time. Oracle Solaris has eight run levels, which are described in the following table. The default run level is specified in the `/etc/inittab` file as run level 3.

Besides using typical boot commands to boot the system to a specified state, the `svcadm` command can be used to change the run level of a system by selecting a milestone at which to run. The following table shows how run levels correspond to milestones.

TABLE 4–2 Oracle Solaris Run Levels

Run Level	Init State	Type	Purpose
0	Power-down state	Power-down	To shut down the operating system so that it is safe to turn off power to the system.
s or S	Single-user state	Single-user	To run as a single user with some file systems mounted and accessible.
1	Administrative state	Single-user	To access all available file systems. User logins are disabled.
2	Multiuser state	Multiuser	For normal operations. Multiple users can access the system and all file systems. All daemons are running except for the NFS server daemons.

TABLE 4–2 Oracle Solaris Run Levels *(Continued)*

Run Level	Init State	Type	Purpose
3	Multiuser level with NFS resources shared	Multiuser	For normal operations with NFS resources shared. This is the default run level.
4	Alternative multiuser state	Multiuser	Not configured by default, but available for customer use.
5	Power-down state	Power-down	To shut down the operating system so that it is safe to turn off power to the system. If possible, automatically turns off power on systems that support this feature.
6	Reboot state	Reboot	To shut down the system to run level 0, and then reboot to a multiuser level with NFS resources shared (or whatever run level is the default in the `inittab` file).

Determining a System's Current Run Level

To determine a system's current run level, use the `who -r` command.

EXAMPLE 4–1 Determining a System's Run Level

The output of the `who -r` command displays information about a system's current run level, as well as previous run levels.

```
$ who -r
  .    run-level 3  Dec 13 10:10  3  0 S
$
```

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

Use this procedure to boot a SPARC based system that is currently at run level 0 to run level 3. For instructions on booting an x86 based system to run level 3, see "Booting an x86 Based System to a Multiuser State (Run Level 3)" in *Booting and Shutting Down Oracle Solaris on x86 Platforms*.

1 Bring the system to the `ok` PROM prompt.

2 Boot the system to run level 3.

```
ok boot
```

The automatic boot procedure displays a series of startup messages and brings the system to run level 3. For more information, see the `boot(1M)` man page.

3 **Verify that the system has booted to run level 3.**

The login prompt is displayed when the boot process has finished successfully.

hostname `console login:`

Example 4–2 SPARC: Booting a System to a Multiuser State (Run Level 3)

The following example shows the messages from booting a system to run level 3.

```
ok boot
Probing system devices
Probing memory
ChassisSerialNumber FN62030249
Probing I/O buses

.
.
.
.
OpenBoot 4.30.4.a, 8192 MB memory installed, Serial #51944031.
Ethernet address 0:3:ba:18:9a:5f, Host ID: 83189a5f.
Rebooting with command: boot
Boot device: /pci@1c,600000/scsi@2/disk@0,0:a  File and args:
SunOS Release 5.11 Version fips_checksum_nightly 64-bit
Copyright (c) 1983, 2011, Oracle and/or its affiliates. All rights reserved.
DEBUG enabled
misc/forthdebug (455673 bytes) loaded
Hardware watchdog enabled
Hostname: portia-123
NIS domain name is solaris.us.oracle.com

portia-123 console login: NIS domain name is solaris.us.oracle.com
```

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

The following procedure describes how to boot an x86 based system to a single-user state (run level S). For instructions on booting a SPARC based system to run level S, see "How to Boot a System to a Single-User State (Run Level S)" in *Booting and Shutting Down Oracle Solaris on SPARC Platforms*

1 **Reboot the system.**

```
# reboot
```

If the system displays the `Press any key to reboot` prompt, press any key to reboot the system.

You can also use the Reset button at this prompt. If the system is shut down, turn the system on with the power switch.

When the boot sequence begins, the GRUB main menu is displayed.

2. **When the GRUB main menu is displayed, type `e` to edit the GRUB menu.**

3. **Depending on the release you are running, use the arrow keys to select the `kernel$` line.**

   If you cannot use the arrow keys, use the caret (^) key to scroll up and the letter v key to scroll down.

4. **Type `e` again to edit the boot entry.**

   From here, you can add options and arguments to the `kernel` or `kernel$` line.

5. **To boot the system to a single-user state, type `-s` at the end of the boot entry line, then press Return to go back to the previous screen.**

   Note – To specify other boot behaviors, replace the `-s` option with the appropriate boot option.

   The following alternate boot behaviors can be specified in this manner:

   - Perform a reconfiguration boot
   - Boot the system with the kernel debugger
   - Redirect the console

   For more information, see the `boot(1M)` man page.

6. **To boot the system to a single-user state, type `b`.**

7. **When prompted, type the `root` password.**

8. **Verify that the system is at run level S.**

   ```
   # who -r
   .       run-level S  Jun 13 11:07     S      0  0
   ```

9. **Perform the system maintenance task that required the run level change to S.**

10. **After you complete the system maintenance task, reboot the system.**

Shutting Down a System (Task Map)

TABLE 4–3 Shutting Down a System: Task Map

Task	Description	For Instructions
Shut down a system by using the `shutdown` command.	Use the `shutdown` command with the appropriate options to shut down a system. This method is preferred for shutting down a server.	"How to Shut Down a System by Using the `shutdown` Command" on page 76
Shut down a system by using the `init` command.	Use the `init` command and indicate the appropriate run level to shut down a system.	"How to Shut Down a System by Using the `init` Command" on page 78

Shutting Down a System

Oracle Solaris is designed to run continuously so that the electronic mail and network software can work correctly. However, some system administration tasks and emergency situations require that the system be shut down to a level where it is safe to remove power. In some cases, the system needs to be brought to an intermediate level, where not all system services are available.

Such cases include the following:

- Adding or removing hardware
- Preparing for an expected power outage
- Performing file system maintenance, such as a backup

For information about using your system's power management features, see the `poweradm`(1M) man page.

For detailed information when to shutdown a system and which commands to use, see "System Shutdown Commands" in *Booting and Shutting Down Oracle Solaris on SPARC Platforms*.

▼ How to Shut Down a System by Using the `shutdown` Command

1 Become the `root` role.

2 For a multiuser timesharing server shutdown, find out if any users are logged in to the system.

```
# who
```

3 Shut down the system.

```
# shutdown -iinit-state -ggrace-period -y
```

`-i`*init-state*	Brings the system to an init state that is different from the default of S. The choices are 0, 1, 2, 5, and 6. Run levels 0 and 5 are states reserved for shutting the system down. Run level 6 reboots the system. Run level 2 is available as a multiuser operating state.
`-g`*grace-period*	Indicates a time (in seconds) before the system is shut down. The default is 60 seconds.
`-y`	Continues to shut down the system without intervention. Otherwise, you are prompted to continue the shutdown process after 60 seconds.

For more information, see the `shutdown`(1M) man page.

4 If you are asked for confirmation, type `y`.

```
Do you want to continue? (y or n): y
```

If you used the `shutdown -y` command, you will not be prompted to continue.

5 Type the `root` password, if prompted.

```
Type Ctrl-d to proceed with normal startup,
(or give root password for system maintenance): xxxxxx
```

6 After you have finished performing any system administration tasks, press Control-D to return to the default system run level.

Example 4–3 SPARC: Bringing a System to a Shutdown State (Run Level 0) by Using the `shutdown` Command

In the following example, the `shutdown` command is used to bring a SPARC based system to run level 0 in five minutes without requiring additional confirmation.

```
# who
root       console       Jun 17 12:39
userabc         pts/4        Jun 17 12:39   (:0.0)
# shutdown -i0 -g300 -y
Shutdown started.    Thu Jun 17 12:40:25...

Broadcast Message from root (console) on pretend Thu Jun 17 12:40:25...
The system pretend will be shut down in 5 minutes
.
.
.
Changing to init state 0 - please wait
#
```

```
INIT: New run level: 0
The system is coming down.  Please wait.
System services are now being stopped.
.
.
.
The system is down.
syncing file systems... done
Program terminated
Type  help  for more information
ok
```

See Also Regardless of why you shut down a system, you will probably want to return to run level 3, where all file resources are available, and users can log in. For instructions on bringing a system back to a multiuser state, "SPARC: How to Boot a System to a Multiuser State (Run Level 3)" on page 73.

Bringing a System to a Shutdown State (Run Level 0) by Using the `init` Command

Run levels 0, 5, and 6 are reserved for shutting down a system. Bringing a system to run level 0 enables power to the system to be safely turned off. As shown in the example that follows, the `init` command is used to bring a system to run level 0.

▼ How to Shut Down a System by Using the `init` Command

Use this procedure when you need to shut down a stand-alone system.

1. **Become the `root` role.**
2. **Shut down the system.**

   ```
   # init 5
   ```

   For more information, see the `init`(1M) man page.

Example 4–4 Bringing a System to the Shutdown State (Run Level 0) by Using the `init` Command

In the following example, the `init` command is used to bring a system to the level where it is safe to turn off power:

```
# init 0
#
INIT: New run level: 0
The system is coming down.  Please wait.
.

.

.
The system is down.
syncing file systems... [11] [10] [3] done
Press any key to reboot
```

Booting a System From the Network

You might need to boot a stand-alone system from the network for recovery purposes, if the system cannot boot from the local disk. Any system can boot from the network, if a boot server is available.

To boot a SPARC based system from the network, a DHCP server is required. Also required is a boot server that provides `tftp` service. The DHCP server supplies the information that the client needs to configure its network interface.

You can boot an x86 based system directly from a network that supports the PXE network boot protocol. The default network boot strategy that is used for both PXE and non-PXE devices is DHCP. If no PXE or DHCP server is available, you can load GRUB from a diskette, a CD-ROM, or local disk.

For SPARC based systems, the process of booting over a local area network (LAN) and booting over a wide area network (WAN) is slightly different. In both network boot scenarios, the PROM downloads the booter from a boot server or an install server, which is `inetboot` in this case.

When booting over a (LAN), the firmware uses DHCP to discover the boot or install server. TFTP is then used to download the booter, which is `inetboot` in this case. When booting over a WAN, the firmware uses either DHCP or NVRAM properties to discover the install server, the router, and the proxies that are required for the system to boot from the network. The protocol that is used to download the booter is HTTP. In addition, the booter's signature might be checked with a predefined private key.

For more information, see the following references:

- For more information about how DHCP works in this Oracle Solaris release, see Part II, "DHCP," in *Oracle Solaris Administration: IP Services*.
- For detailed instructions on booting an x86 system from the network, see Chapter 5, "Booting an x86 Based System From the Network (Tasks)," in *Booting and Shutting Down Oracle Solaris on x86 Platforms*.

- For detailed instructions on booting a SPARC based system from the network, see Chapter 5, "Booting a SPARC Based System From the Network (Tasks)," in *Booting and Shutting Down Oracle Solaris on SPARC Platforms*.
- For detailed information about setting up an install server, an install client, and other installation options, see *Installing Oracle Solaris 11 Systems*.

Accelerating the Reboot Process (Task Map)

TABLE 4–4 Accelerating the Reboot Process: Task Map

Task	Description	For Instructions
Initiate a fast reboot of a SPARC based system.	On SPARC based systems that do not have the Fast Reboot feature enabled, use the `reboot` command with the `-f` option. If the Fast Reboot feature has been enabled, you can use either the `reboot` or the `init 6` command to automatically initiate a fast reboot of a SPARC based system.	"How to Initiate a Fast Reboot of a SPARC Based System" on page 81
Initiate a fast reboot of an x86 based system, bypassing the BIOS.	Because Fast Reboot is the default boot mode in this release, you can use either the `reboot` or the `init 6` command to initiate a fast reboot of the system.	"How to Initiate a Fast Reboot of an x86 Based System" on page 82
Change the default behavior of the Fast Reboot feature.	On x86 based systems, the Fast Reboot feature is enabled by default. On SPARC based systems, the Fast Reboot feature is supported, but *not* enabled by default. You can configure the default behavior of the Fast Reboot feature on a SPARC based system so that a fast reboot is initiated by default.	"Changing the Default Behavior of the Fast Reboot Feature" on page 82
Initiate a standard reboot of a system that has Fast Reboot enabled.	Use the `reboot` command with the `-p` option to perform a standard reboot of the system that has the Fast Reboot feature enabled.	"Initiating a Standard Reboot of a System That Has Fast Reboot Enabled" on page 83

Accelerating the Reboot Process

The Fast Reboot feature of Oracle Solaris is supported on both SPARC and x86 platforms. The Fast Reboot feature behaves differently on SPARC based systems than it does on an x86 based systems. On x86 based systems, Fast Reboot is the default. On SPARC based systems, the behavior is supported, but to initiate a fast reboot of a system, you must specify the `-f` option with the `reboot` command.

On a SPARC based system, using the `-f` option with the `reboot` command accelerates the boot process and skips certain POST tests. On an x86 based system, the feature is enabled by default, which means you do not have to use the `-f` option with the `reboot` command to initiate a fast reboot of the system.

Note – Fast reboot on SPARC is applicable only to certain system types. On `sun4v` systems, fast reboot is unnecessary because the reboot is actually a hypervisor restart that does not involve POST.

The Fast Reboot feature is implemented through the boot configuration SMF service, `svc:/system/boot-config`. This service provides a means for setting or changing default boot configuration properties. When the `config/fastreboot_default` `property` is set to `true`, the system automatically performs a fast reboot, without the need to use the `reboot -f` command. By default, this property is set to `false` on SPARC platforms. For instructions on making a fast reboot the default behavior on a SPARC based system, see "Changing the Default Behavior of the Fast Reboot Feature" on page 82.

Note – On SPARC based systems the `boot-config` service also requires the `solaris.system.shutdown` authorization as the `action_authorization` and `value_authorization`.

▼ How to Initiate a Fast Reboot of a SPARC Based System

Use the following procedure to initiate a fast reboot of a SPARC based system when the `config/fastreboot_default` property of the `boot-config` service is set to `false`, which is the default behavior. To change the default behavior of the Fast Reboot feature so that a fast reboot is automatically initiated when the system reboots, see "Changing the Default Behavior of the Fast Reboot Feature" on page 82.

1. **Become the `root` role.**

2. **Initiate a fast reboot of the system by typing the following command:**

```
# reboot -f
```

▼ How to Initiate a Fast Reboot of an x86 Based System

Note – In this Oracle Solaris release, Fast Reboot is the default operating mode on x86 based systems. Previously, to initiate a fast reboot of an x86 based system, you needed to specify the `-f` option with the `reboot` command to initiate a fast reboot of the system. You no longer need to specify this option.

1. **Become the `root` role.**
2. **To initiate a fast reboot of the system, type either of the following commands:**

```
# reboot

# init 6
```

Changing the Default Behavior of the Fast Reboot Feature

The `config/fastreboot_default` property of the `boot-config` service enables an automatic fast reboot of the system when either the `reboot` or the `init 6` command is used. When the `config/fastreboot_default` property is set to `true`, the system automatically performs a fast reboot, without the need to use the `reboot -f` command. By default, this property's value is set to `false` on a SPARC based system and `true` on an x86 based system.

To configure the properties that are part of the `boot-config` service use the `svccfg` and `svcadm` commands.

For example, to set the property's value to `true` (enabled) on a SPARC based system, type the following commands:

```
# svccfg -s "system/boot-config:default" setprop config/fastreboot_default=true
# svcadm refresh svc:/system/boot-config:default
```

Setting the property's value to `true` enables the fast reboot process, which bypasses certain POST tests. When this property is set to `true`, you do not have to use the `-f` option with the `reboot` command to initiate a fast reboot of the system.

For information about managing the boot configuration service through SMF, see the `svcadm`(1M) and `svccfg`(1M) man pages.

Initiating a Standard Reboot of a System That Has Fast Reboot Enabled

To reboot a system that has the Fast Reboot feature enabled, without having to reconfigure the properties of the `boot-config` service, use the `-p` option with the `reboot` command, as follows:

```
# reboot -p
```

For more information about rebooting a SPARC based system, see Chapter 4, "Rebooting a SPARC Based System (Tasks)," in *Booting and Shutting Down Oracle Solaris on SPARC Platforms*.

For more information about rebooting an x86 based system, see Chapter 4, "Rebooting an x86 Based System (Tasks)," in *Booting and Shutting Down Oracle Solaris on x86 Platforms*.

Booting From a ZFS Boot Environment (Task Map)

The following procedures describe how to boot from a ZFS boot environment or root file system on the SPARC and x86 platforms.

For detailed information about managing boot environments, see *Creating and Administering Oracle Solaris 11 Boot Environments*.

TABLE 4–5 Booting From a ZFS Boot Environment: Task Map

Task	Description	For Instructions
Display a list of boot environments and datasets during the boot sequence on a SPARC based system.	To display a list of boot environments that are on a system during the boot sequence, specify the `-L` option with the `boot` command.	"SPARC: How to Display a List of Available Boot Environments During the Boot Sequence" on page 85
Boot from a specified boot environment, dataset, or root file system on a SPARC based system.	Use the `boot -Z` option to boot a specified ZFS boot environment, snapshot, or dataset. **Note –** This option is only supported for boot devices that contain a ZFS pool.	"SPARC: How to Boot From a ZFS Boot Environment or Root File System" on page 85

SPARC: Booting From a ZFS Boot Environment

On SPARC platforms, the following two options of the `boot` command support booting from a ZFS boot environment or root file system:

`-L` Displays a list of available boot environments within a ZFS pool.

Note – The boot `-L` command is executed from the OBP, *not* from the command line.

`-Z` *dataset* Boots the root file system for the specified ZFS boot environment.

If you are booting a system from a ZFS root file system, first use the `boot` command with the `-L` option from the OBP to print a list of the available boot environments on the system. Then, use the `-Z` option to boot the specified boot environment.

For more information, see the `boot(1M)` man page.

On x86 platforms, the following entries are added to the */pool-name*`/boot/grub/menu.lst` file during the installation process or during the `beadm activate` operation to boot ZFS automatically:

```
title 2010-12-10-be-s
findroot (pool_rpool,0,a)
bootfs rpool/ROOT/2010-12-10-be_152
kernel$ /platform/i86pc/kernel/$ISADIR/unix -B $ZFS-BOOTFS -s
module$ /platform/i86pc/$ISADIR/boot_archive
```

If the device that is identified by GRUB as the boot device contains a ZFS storage pool, the `menu.lst` file is used to create the GRUB menu. On an x86 based system with multiple ZFS boot environments, you can select a boot environment from the GRUB menu during boot time. If the root file system that corresponds to this menu entry is a ZFS dataset, the following option is added:

```
-B $ZFS-BOOTFS
```

The `$ZFS-BOOTFS` keyword enables you to boot from an Oracle Solaris ZFS root file system on an x86 based system. This option identifies which boot environment or dataset to boot. If you install an Oracle Solaris release that supports a ZFS boot loader, the GRUB `menu.lst` file, as well as the GRUB boot menu, contains this information by default.

For more information about booting from a ZFS boot environment or root file system, see "Booting From a ZFS Boot Environment or Root File System on x86 Platforms" in *Booting and Shutting Down Oracle Solaris on x86 Platforms*.

SPARC: How to Display a List of Available Boot Environments During the Boot Sequence

On SPARC based systems, the `menu.lst` file contains the following two commands:

- `title` – Provides a title for a boot environment
- `bootfs` – Specifies the full name of the boot environment

As explained in the following procedure, to display a list of the boot environments within a ZFS pool, use the `boot -L` command. This command displays a list of the available boot environments within a given ZFS root pool and provides instructions for booting the system.

1. **Become the `root` role.**
2. **Bring the system to the `ok` PROM prompt.**

   ```
   # init 0
   ```

3. **List the available boot environments in a ZFS pool.**

   `ok` **`boot`** *device-specifier* **`-L`**

   where *device-specifier* identifies a storage pool, *not* a single root file system.

4. **To boot one of the entries that is displayed, type the number that corresponds to the entry.**
5. **Boot the specified boot environment by following the instructions that are displayed on the screen.**

   For instructions, see "SPARC: How to Boot From a ZFS Boot Environment or Root File System" on page 85.

See Also For more information, see Chapter 5, "Managing ZFS Root Pool Components," in *Oracle Solaris Administration: ZFS File Systems*.

SPARC: How to Boot From a ZFS Boot Environment or Root File System

When booting from ZFS, the *device-specifier* identifies a storage pool, *not* a single root file system. A storage pool can contain multiple boot environments, datasets, or root file systems. Therefore, when booting from ZFS, you must also identify a root file system within the pool that is identified by the boot device as the default. The default boot device is identified by the pool's `bootfs` property. This procedure shows how to boot the system by specifying a ZFS boot environment. See the `boot(1M)` man page for a complete description of all the boot options that are available.

Note – In Oracle Solaris 11, a ZFS root file system is booted by default. Use this procedure to specify a ZFS root file system from which to boot.

For more information, see the `zpool(1M)` man page.

1. **Become the `root` role.**

2. **Bring the system to the `ok` PROM prompt.**

   ```
   # init 0
   ```

3. **(Optional) Display a list of available boot environments by using the `boot` command with the `-L` option.**

   For instructions, see "SPARC: How to Display a List of Available Boot Environments During the Boot Sequence" on page 85.

4. **To boot a specified entry, type the number of the entry and press Return:**

   ```
   Select environment to boot: [1 - 2]:
   ```

5. **To boot the system, follow the instructions that are displayed on the screen.**

   ```
   To boot the selected entry, invoke:
   boot [<root-device>] -Z rpool/ROOT/boot-environment

   ok boot -Z rpool/ROOT/boot-environment
   ```

   For example:

   ```
   # boot -Z rpool/ROOT/zfs2BE
   ```

6. **After the system has booted, verify the active boot environment.**

   ```
   # prtconf -vp | grep whoami
   ```

7. **(Optional) To display the boot path for the active boot environment, type the following command:**

   ```
   # prtconf -vp | grep bootpath
   ```

8. **(Optional) To determine whether the correct boot environment was booted, type the following command:**

   ```
   # df -lk
   ```

Example 4–5 SPARC: Booting From a ZFS Boot Environment

This example shows how to use the `boot -Z` command to boot a ZFS boot environment on a SPARC based system.

```
# init 0
# svc.startd: The system is coming down. Please wait.
svc.startd: 79 system services are now being stopped.
svc.startd: The system is down.
syncing file systems... done
Program terminated
ok boot -Z rpool/ROOT/zfs2BEe
Resetting
LOM event: =44d+21h38m12s host reset
g ...

rProcessor Speed = 648 MHz
Baud rate is 9600
8 Data bits, 1 stop bits, no parity (configured from lom)

.
.
.
Environment monitoring: disabled
Executng last command: boot -Z rpool/ROOT/zfs2BE
Boot device: /pci@1f,0/pci@1/scsi@8/disk@0,0 File and args: -Z rpool/ROOT/zfs2Be
zfs-file-system
.
.
.
Hostname: mallory
NIS domainname is ...
Reading ZFS config: done.
Mounting ZFS filesytems: (6/6)

mallory console login:
```

See Also For more information about booting from a ZFS root file system, see "Booting From a ZFS Root File System" in *Oracle Solaris Administration: ZFS File Systems*.

Modifying Boot Parameters (Task Map)

TABLE 4–6 Modifying Boot Parameters: Task Map

Task	Description	For Instructions
Determine the current boot device on a SPARC based system.	Use this procedure to determine the current default boot device from which the system will boot.	"SPARC: How to Determine the Default Boot Device" on page 89

TABLE 4–6 Modifying Boot Parameters: Task Map *(Continued)*

Task	Description	For Instructions
Change the default boot device on a SPARC based system.	To change the default boot device, use one of the following methods: ■ Change the `boot-device` parameter at `ok` PROM prompt. ■ Change the `boot-device` parameter by using the `eeprom` command.	"SPARC: How to Change the Default Boot Device by Using the Boot PROM" on page 90 "How to Change the Default Boot File by Using the eeprom Utility" in *Booting and Shutting Down Oracle Solaris on SPARC Platforms*
Modify boot parameters on an x86 based system by using the `eeprom` command.	Modify boot parameters on an x86 based system by using the `eeprom` command. Boot parameters that are set by using the `eeprom` command persist over a system reboot, unless these options are overridden by editing the GRUB menu at boot time.	"x86: How to Modify Boot Parameters by Using the `eeprom` Command" on page 91
Modify boot parameters on an x86 based system by editing the GRUB menu at boot time.	Boot options that are specified by editing the GRUB menu at boot time only persist until the next time the system is booted.	"x86: How to Modify Boot Parameters at Boot Time" on page 92
Modify boot behavior on an x86 based system by editing the `menu.lst` configuration file.	Use this method to add new OS entries or redirect the console. Changes that you make to the file persist over system reboots.	"Modifying Boot Entries and Parameters by Editing the menu.lst File" in *Booting and Shutting Down Oracle Solaris on x86 Platforms*

Modifying Boot Parameters

On SPARC platforms, the boot PROM is used to boot a SPARC based system and to modify boot parameters. For example, you might want to reset the device from which to boot, change the default boot file or kernel, or run hardware diagnostics before bringing the system to a multiuser state.

For a complete list of PROM commands, see the `monitor(1M)` and `eeprom(1M)` man pages.

On x86 platforms, the primary methods for modifying boot parameters are as follows:

- By using the `eeprom` command

  The `eeprom` command is used to assign a different value to a standard set of parameters. These values, which are equivalent to the SPARC OpenBoot PROM NVRAM variables, are stored either in the `/boot/solaris/bootenv.rc` file or in the `menu.lst` file. Changes that

are made to boot parameters by using the `eeprom` command persist over each system reboot and are preserved during a software upgrade. See the `eeprom(1M)` man page for more information.

Caution – If you directly edit the `menu.lst` file, certain boot parameters (`boot-file`, `boot-arguments`, and `console`) cannot be changed at a later time by using the `eeprom` command.

- By editing the GRUB menu at boot time

  Changes that are made by modifying the default kernel usage at boot time override options that you set by using the `eeprom` command. However, these changes only remain in effect until the next time you boot the system. See the `kernel(1M)` man page for more information.
- By editing the GRUB configuration file (`menu.lst`)

▼ SPARC: How to Determine the Default Boot Device

1. **Bring the system to the `ok` PROM prompt.**

   For more information, see "How to Shut Down a System by Using the `init` Command" on page 78.

2. **Determine the default boot device.**

   ```
   ok printenv boot-device
   ```

   `boot-device` Identifies the parameter for setting the device from which to boot.

   For more information, see the `printenv(1B)` man page.

   The default `boot-device` is displayed in a format that is similar to the following:

   ```
   boot-device = /pci@1f,4000/scsi@3/disk@1,0:a
   ```

   If the `boot-device` parameter specifies a network boot device, the output is similar to the following:

   ```
   boot-device = /sbus@1f,0/SUNW,fas@e,8800000/sd@a,0:a \
   /sbus@1f,0/SUNW,fas@e,8800000/sd@0,0:a disk net
   ```

▼ SPARC: How to Change the Default Boot Device by Using the Boot PROM

Before You Begin You might need to identify the devices on the system before you can change the default boot device to some other device. For information about identifying devices on the system, see "How to Identify Devices on a System" in *Booting and Shutting Down Oracle Solaris on SPARC Platforms*.

1 Bring the system to the ok PROM prompt.

```
# init 0
```

2 Change the value of the `boot-device` parameter.

```
ok setenv boot-device device[n]
```

`device[n]` Identifies the `boot-device` value, such as `disk` or `network`. The *n* can be specified as a disk number. Use one of the `probe` commands if you need help identifying the disk number.

3 Verify that the default boot device has been changed.

```
ok printenv boot-device
```

4 Save the new `boot-device` value.

```
ok reset-all
```

The new `boot-device` value is written to the PROM.

Example 4–6 SPARC: Changing the Default Boot Device by Using the Boot PROM

In this example, the default boot device is set to disk.

```
# init 0
#
INIT: New run level: 0
.
.
.
The system is down.
syncing file systems... done
Program terminated
ok setenv boot-device /pci@1f,4000/scsi@3/disk@1,0
boot-device =         /pci@1f,4000/scsi@3/disk@1,0
ok printenv boot-device
boot-device           /pci@1f,4000/scsi@3/disk@1,0
ok boot
Resetting ...

screen not found.
Can't open input device.
```

```
Keyboard not present.  Using ttya for input and output.
.
.
.
Rebooting with command: boot disk1
Boot device: /pci@1f,4000/scsi@3/disk@1,0  File and args:
```

In this example, the default boot device is set to the network.

```
# init 0
#
INIT: New run level: 0
.
.
.
The system is down.
syncing file systems... done
Program terminated
ok setenv boot-device net
boot-device =         net
ok printenv boot-device
boot-device           net                   disk
ok reset
.
.
.
Boot device: net  File and args:

pluto console login:
```

See Also For instructions on using the `eeprom` utility to change the default boot device on a SPARC based system, see "How to Change the Default Boot Device by Using the eeprom Utility" in *Booting and Shutting Down Oracle Solaris on SPARC Platforms*.

▼ x86: How to Modify Boot Parameters by Using the `eeprom` Command

1. **Become the `root` role.**
2. **Change the specified parameter.**

   `# eeprom` *parameter*`=`*new-value*
3. **Verify that the new parameter has been set.**

   `# eeprom` *parameter*

   The output should display the new `eeprom` value for the specified parameter.

Example 4–7 Setting the `auto-boot` Parameter by Using the `eeprom` Command

The following example shows how to set the `auto-boot` boot parameter to true.

```
# eeprom auto-boot?=true
```

When the `eeprom` command is run in user mode, any parameters that have a trailing question mark (`?`) need to be enclosed in double quotation marks to prevent the shell from interpreting the question mark. Preceding the question mark with an escape character (`\`) also prevents the shell from interpreting the question mark. For example:

```
# eeprom "auto-boot?"=true
```

▼ x86: How to Modify Boot Parameters at Boot Time

When you modify the default kernel usage by editing the GRUB menu at boot time, the changes do not persist over a system reboot. The default boot parameters are restored the next time you boot the system.

For a detailed description of all of the boot parameters that you can specify at boot time, see "Modifying Based Boot Parameters at Boot Time" in *Booting and Shutting Down Oracle Solaris on x86 Platforms*.

1. **Reboot the system.**

   When the boot sequence begins, the GRUB main menu is displayed.

2. **Use the arrow keys to select the boot entry to edit.**

3. **Type `e` to access the GRUB edit menu.**

4. **Select the `kernel$` line in the menu.**

5. **Type `e` to add boot arguments to the line.**

6. **Type any additional boot arguments.**

7. **Press Return to save your changes and return to the previous menu.**

   Note – Pressing the Escape key returns you to the GRUB main menu without saving your changes.

8. **To boot the system, type `b`.**

   Changes you make take effect when the system is booted.

Adding a Linux Entry to the GRUB Menu After an Installation

If you are setting up a boot environment in such a way that you install Linux on one partition first and Oracle Solaris on another partition afterwards, you will need to follow certain instructions to ensure that the GRUB menu information from the new installation does not erase the GRUB menu information from a previous installation. For instructions, see "How to Add a Linux Entry to the GRUB Menu After Installing Oracle Solaris" in *Booting and Shutting Down Oracle Solaris on x86 Platforms*.

Note – Some Linux distributions now run on GRUB2, for example, Ubuntu and Mint Linux. You cannot boot GRUB2 partitions on the version of GRUB that is included in Oracle Solaris 11. In these instances, an alternate workaround is suggested.

Keeping a System Bootable (Task Map)

TABLE 4–7 Keeping a System Bootable: Task Map

Task	Description	For Instructions
Determine whether the `boot-archive` service is running.	The `boot-archive` service is controlled by SMF. Use the `svcs` command to verify whether the `boot-archive` service is running. Use the `svcadm` command to enable or disable the service.	"Determining Whether the `boot-archive` SMF Service Is Running" on page 94
Clear a boot archive update failure by using the `bootadm` command to manually update the boot archive.	Use this procedure to manually clear boot archive update failures.	"How to Clear a Failed Automatic Boot Archive Update by Manually Updating the Boot Archive" on page 95
Clear a boot archive update failure on an x86 based system by using the `auto-reboot-safe` property.	Use this procedure in cases where the boot archive update on an x86 based system fails because the `auto-reboot-safe` property is set to `false`.	"x86: How to Clear a Failed Automatic Boot Archive Update by Using the `auto-reboot-safe` Property" on page 95

Keeping a System Bootable

The `bootadm` command handles the details of boot archive update and verification. During the process of a normal system shutdown, the shutdown process compares the boot archive's contents with the root file system. If there have been updates to the system such as drivers or configuration files, the boot archive is rebuilt to include these changes so that upon reboot, the boot archive and root file system are synchronized.

The files in the x86 boot archive are located in the `/platform/i86pc/amd64/boot_archive` directory.

The files in the SPARC boot archive are located in the `/platform` directory.

To list the contents of the boot archive by using the `bootadm list-archive` command, as follows:

```
# bootadm list-archive
```

Whenever any files in the boot archive are updated, the archive must be rebuilt. For modifications to take effect, the rebuild of the archive must take place before the next system reboot.

Determining Whether the `boot-archive` SMF Service Is Running

If the `boot-archive` service is disabled, automatic recovery of the boot archives upon a system reboot might not occur. As a result, the boot archives could become unsynchronized or corrupted, preventing the system from booting.

To determine whether the `boot-archive` service is running, use the `svcs` command, as follows:

```
$ svcs boot-archive
STATE          STIME    FMRI
online         Mar_31   svc:/system/boot-archive:default
```

To enable or disable the `boot-archive` service, type:

```
# svcadm enable | disable system/boot-archive
```

To verify the state of the `boot-archive` service, type:

```
# svcs boot-archive
```

If the service is running, the output displays an online service state.

How to Clear a Failed Automatic Boot Archive Update by Manually Updating the Boot Archive

During the process of booting the system, if a warning message that is similar to the following is displayed, take action accordingly:

```
WARNING: Automatic update of the boot archive failed.
Update the archives using 'bootadm update-archive'
command and then reboot the system from the same device that
was previously booted.
```

The following procedure describes how to manually update an out-of-date boot archive by using the `bootadm` command.

Note – The same procedure can also be used to manually update the boot archive.

---

1. **Become the `root` role.**

2. **To update the boot archive, type the following command:**

   ```
   # bootadm update-archive
   ```

Note – To update the boot archive on an alternate root, type the following command:

```
# bootadm update-archive -R /a
```

`-R` *altroot* Specifies an alternate `root` path to apply to the `update-archive` subcommand.

Caution – The root file system of any non-global zone must not be referenced with the `-R` option. Doing so might damage the global zone's file system, compromise the security of the global zone, or damage the non-global zone's file system. See the `zones(5)` man page.

---

3. **Reboot the system.**

   ```
   # reboot
   ```

x86: How to Clear a Failed Automatic Boot Archive Update by Using the `auto-reboot-safe` Property

Boot archive recovery on x86 platforms is automated through the Fast Reboot feature. However, during the process of booting the system, if a warning similar to the following is displayed:

```
WARNING: Reboot required.
The system has updated the cache of files (boot archive) that is used
during the early boot sequence. To avoid booting and running the system
with the previously out-of-sync version of these files, reboot the
system from the same device that was previously booted.
```

The system then enters system maintenance mode. As a result, the automatic update of the boot archive fails. To correct the problem, follow the steps in this procedure.

1. **Become the `root` role.**

2. **Reboot the system.**

   ```
   # reboot
   ```

3. **If the active BIOS boot device and the GRUB menu entries point to the current boot instance, follow these steps to prevent a boot archive update failure:**

   a. **Set the `auto-reboot-safe` property of the `svc:/system/boot-config` SMF service to `true`, as follows:**

      ```
      # svccfg -s svc:/system/boot-config:default setprop config/auto-reboot-safe = true
      ```

   b. **Verify that the `auto-reboot-safe` property is set correctly.**

      ```
      # svccfg -s svc:/system/boot-config:default listprop |grep config/auto-reboot-safe
      config/auto-reboot-safe            boolean  true
      ```

Where to Find More Information About Booting and Shutting Down a System

TABLE 4–8 Booting and Shutdown Tasks

Task	SPARC Information	x86 Information
Detailed overview of booting a system.	Chapter 1, "Booting and Shutting Down a SPARC Based System (Overview)," in *Booting and Shutting Down Oracle Solaris on SPARC Platforms*	Chapter 1, "Booting and Shutting Down an x86 Based System (Overview)," in *Booting and Shutting Down Oracle Solaris on x86 Platforms*
Bring a system to a specified state (run level booting).	Chapter 2, "Booting a SPARC Based System to a Specified State (Tasks)," in *Booting and Shutting Down Oracle Solaris on SPARC Platforms*	Chapter 2, "Booting an x86 Based System to a Specified State (Tasks)," in *Booting and Shutting Down Oracle Solaris on x86 Platforms*

TABLE 4–8 Booting and Shutdown Tasks *(Continued)*

Task	SPARC Information	x86 Information
Shut down a system.	Chapter 3, "Shutting Down a System (Tasks)," in *Booting and Shutting Down Oracle Solaris on SPARC Platforms*	Chapter 3, "Shutting Down a System (Tasks)," in *Booting and Shutting Down Oracle Solaris on x86 Platforms*
Reboot a system.	Chapter 4, "Rebooting a SPARC Based System (Tasks)," in *Booting and Shutting Down Oracle Solaris on SPARC Platforms*	Chapter 4, "Rebooting an x86 Based System (Tasks)," in *Booting and Shutting Down Oracle Solaris on x86 Platforms*
Boot a system from the network.	Chapter 5, "Booting a SPARC Based System From the Network (Tasks)," in *Booting and Shutting Down Oracle Solaris on SPARC Platforms*	Chapter 5, "Booting an x86 Based System From the Network (Tasks)," in *Booting and Shutting Down Oracle Solaris on x86 Platforms*
Modify boot parameters on a system.	Chapter 6, "Modifying Boot Parameters on a SPARC Based System (Tasks)," in *Booting and Shutting Down Oracle Solaris on SPARC Platforms*	Chapter 6, "Modifying Boot Parameters on an x86 Based System (Tasks)," in *Booting and Shutting Down Oracle Solaris on x86 Platforms*
Boot from a specified ZFS boot environment or root file system.	Chapter 7, "Creating, Administering, and Booting From ZFS Boot Environments on SPARC Platforms (Tasks)," in *Booting and Shutting Down Oracle Solaris on SPARC Platforms*	Chapter 7, "Creating, Administering, and Booting From ZFS Boot Environments on x86 Platforms (Tasks)," in *Booting and Shutting Down Oracle Solaris on x86 Platforms*
Keep a system bootable by using the boot administration interface (`bootadm`)	Chapter 8, "Keeping a SPARC Based System Bootable (Tasks)," in *Booting and Shutting Down Oracle Solaris on SPARC Platforms*	Chapter 8, "Keeping an x86 Based System Bootable (Tasks)," in *Booting and Shutting Down Oracle Solaris on x86 Platforms*
Troubleshoot booting a system.	Chapter 9, "Troubleshooting Booting a SPARC Based System (Tasks)," in *Booting and Shutting Down Oracle Solaris on SPARC Platforms*	Chapter 9, "Troubleshooting Booting an x86 Based System (Tasks)," in *Booting and Shutting Down Oracle Solaris on x86 Platforms*

◆ ◆ ◆ CHAPTER 5

Working With Oracle Configuration Manager

This chapter provides an overview of Oracle Configuration Manager, as well as instructions for using the service on a system running an Oracle Solaris release. The following is a list of information that is in this chapter:

- "Introduction to Oracle Configuration Manager" on page 99
- "Managing Oracle Configuration Manager (Tasks)" on page 100

Introduction to Oracle Configuration Manager

Oracle Configuration Manager is used to collect configuration information and upload it to the Oracle repository. Customer support representatives can use this information to provide better service. Some of the benefits of using Oracle Configuration Manager are as follows:

- Reduces time for the resolution of support issues
- Provides proactive problem avoidance
- Improves access to best practices and the Oracle knowledge base
- Improves understanding of customer business needs and provides consistent responses and services

Oracle Configuration Manager can be run in two modes: connected or disconnected. The disconnected mode is needed only if your server does not have a connection to the Internet, and you cannot configure an Oracle Support Hub. In this mode, you can manually collect configuration information and upload the information to Oracle by way of a service request.

In the connected mode, Oracle Configuration Manager can be run in several network configurations as follows:

- Systems can be directly connected to the Internet.
- Systems can be connected to the Internet through a proxy server.

- Systems do not have direct access to the Internet, but they do have access to an intranet proxy server, which in turn has an Internet connection through an Oracle Support Hub.
- Systems do not have direct access to the Internet, but they do have access to an Oracle Support Hub, which in turn is connected to the Internet through a proxy server.

For more information about setting up and configuring Oracle Configuration Manager, see: Oracle Configuration Manager Installation and Administration Guide. The rest of this document focuses on the Oracle Solaris specific tasks that are associated with Oracle Configuration Manager.

Note – To configure Oracle Configuration Manager to use a proxy or an Oracle Support Hub, you must run the `configCCR` command in interactive mode. See Oracle Support Hub for more information.

During an Oracle Solaris 11 installation, the software attempts to set up an anonymous connection to the Oracle repository. If successful, this connection allows the installation process to proceed without prompting for any information. Ideally, you should change the registration or the network configuration after the server is fully installed. Data loaded anonymously is not tied to any organization. If the software could not connect to the Oracle repository, you can register manually, then enable the Oracle Configuration Manager service.

Managing Oracle Configuration Manager (Tasks)

The following task map includes several procedures that are associated with using Oracle Configuration Manager on a Oracle Solaris system. Each row includes a task, a description of when you would want to perform that task, followed by a link to the task.

Task	Description	For Instructions
Enable the Oracle Configuration Manager service.	Enables the Oracle Configuration Manager service, after you have made configuration changes.	"How to Enable the Oracle Configuration Manager Service" on page 101
Disable the Oracle Configuration Manager service.	Disables the Oracle Configuration Manager service, before you make any significant configuration changes.	"How to Disable the Oracle Configuration Manager Service" on page 101
Manually register with the Oracle repository.	Changes your registration credentials.	"How to Manually Register With the Oracle Repository" on page 101
Change data collection time.	Resets the data collection frequency and time.	"How to Change the Time or Frequency of Data Collection" on page 102

How to Enable the Oracle Configuration Manager Service

1. **Become an administrator.**

   For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2. **Enable the Oracle Configuration Manager service.**

   ```
   # svcadm enable system/ocm
   ```

How to Disable the Oracle Configuration Manager Service

1. **Become an administrator.**

   For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2. **Disable the Oracle Configuration Manager service.**

   ```
   # svcadm disable system/ocm
   ```

Caution – Do not run the `emCCR stop` command on an Oracle Solaris system. Any changes to the service must be made using Service Management Facility (SMF).

How to Manually Register With the Oracle Repository

1. **Become an administrator.**

   For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2. **Change user registration.**

   ```
   # configCCR
   ```

   The software prompts for an email account and password. Preferably, use an email account associated with your My Oracle Support identity.

   If the system can communicate directly with the registration server, it does so. If not, you are prompted for the URL of an Oracle Support Hub. If a URL is usable at your site, specify it here. If you do not specify the address of an Oracle Support Hub or still are unable to communicate with the registration server, then you are prompted for a network proxy.

After registration is complete, data collection begins.

See Also For more information about the `configCCR` command, see the `configCCR`(1M) man page or Oracle Configuration Manager Installation and Administration Guide. For complete examples of an interactive session using the `configCCR` command, see configCCR.

▼ How to Change the Time or Frequency of Data Collection

1. **Become an administrator.**

   For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2. **Reset the frequency of data collection.**

   This example resets the collection time to occur weekly on Monday mornings at 6:00 a.m.

   ```
   # emCCR set collection_interval=FREQ=WEEKLY\; BYDAY=MON\; BYHOUR=6
   ```

See Also For more information about the `configCCR` command, see the `emCCR`(1M) man page or Oracle Configuration Manager Installation and Administration Guide.

◆ ◆ ◆ CHAPTER 6

Managing Services (Overview)

This chapter provides an overview of the Oracle Solaris Service Management Facility (SMF) feature. In addition, information about run levels is provided.

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

- "Introduction to SMF" on page 103
- "SMF Concepts" on page 104
- "SMF Administrative and Programming Interfaces" on page 110
- "SMF Components" on page 111
- "SMF Compatibility" on page 115
- "Run Levels" on page 115
- "`/etc/inittab` File" on page 117

For information about the procedures associated with SMF, see "Monitoring Services (Task Map)" on page 119. For information about the procedures associated with run levels, see "Using Run Control Scripts (Task Map)" on page 135.

Introduction to SMF

SMF provides an infrastructure that augments the traditional UNIX startup scripts, `init` run levels, and configuration files. SMF provides a mechanism to define the relationships between applications or services, so that dependent services can automatically be restarted when necessary. Information needed to manage each service is stored in the service repository, which provides a simplified way to manage each service.

SMF defines a set of actions that can be invoked on a service by an administrator. These actions, which can be manually manipulated by the `svcadm` command, include enable, disable, refresh, restart, and mark. Each service is managed by a service restarter, which carries out the administrative actions. In general, the restarters carry out actions by executing methods for a service. Methods for each service are defined in the service configuration repository. These methods allow the restarter to move the service from one state to another state.

The service configuration repository provides a per-service snapshot at the time that each service is successfully started so that fallback is possible. In addition, the repository provides a consistent and persistent way to enable or disable a service, as well as a consistent view of service states. This capability helps you debug service configuration problems.

SMF Concepts

This section presents terms and their definitions within the SMF framework. These terms are used throughout the documentation. To grasp SMF concepts, an understanding of these terms is essential.

SMF Service

The fundamental unit of administration in the SMF framework is the *service instance*. Each SMF service has the potential to have multiple versions of it configured. As well, multiple instances of the same version can run on a single system. An *instance* is a specific configuration of a service. A web server is a service. A specific web server daemon that is configured to listen on port 80 is an instance. Each instance of the web server service could have different configuration requirements. The service has system-wide configuration requirements, but each instance can override specific requirements, as needed. Multiple instances of a single service are managed as child objects of the service object.

Services are not just the representation for standard long-running system services such as `in.dhcpd` or `nfsd`. Services also represent varied system entities that include ISV applications. In addition, a service can represent less traditional entities such as the following:

- A physical network device
- A configured IP address
- Kernel configuration information
- Milestones that correspond to a system init state, such as the multiuser run level

Generically, a service is an entity that provides a list of capabilities to applications and other services, local and remote. A service is dependent on an implicitly and explicitly declared list of local services.

A *milestone* is a special type of service. Milestone services represent a level of system readiness. For example, run levels are represented by milestones in SMF. Also, milestones can be used to indicate the readiness of a group of services, such as `svc:/milestone/name-services:default` for the name services or `svc:/milestone/config:default` for the `sysconfig` service.

SMF Dependencies

Dependencies define the relationships between services. These relationships provide precise fault containment by restarting only those services that are directly affected by a fault, rather than restarting all of the services. Dependencies also provide a scalable and reproducible initialization process. Finally, the definition of precise dependencies allows system startup to take advantage of modern, highly parallel machines because all independent services can be started in parallel.

The restart behavior of a service is defined by the `restart_on` attribute for each dependency. A service can be configured to stop, if the service it is dependent on stops due to an error or for another reason, or is refreshed. After a service is stopped by this process, it will automatically be restarted as soon as the service it is dependent on starts. For example, the `ssh` service has a dependency on the `network/ipfilter` service. The `restart_on` attribute is set to `error`, which means that the `ssh` service will be stopped and automatically restarted if the `network/ipfilter` service stops due to an error. The `ssh` service will not be stopped if the other event types are encountered.

Service Identifiers

Each service instance is named with a Fault Management Resource Identifier or FMRI. The FMRI includes the service name and the instance name. For example, the FMRI for the `rlogin` service is `svc:/network/login:rlogin`, where `network/login` identifies the service and `rlogin` identifies the service instance.

Equivalent formats for an FMRI are as follows:

- `svc://localhost/system/system-log:default`
- `svc:/system/system-log:default`
- `system/system-log:default`

In addition, many SMF commands may use an abbreviated service or instance name, when there is no ambiguity. For example, `system-log` can be used directly rather than the longer formats. See the SMF command man pages, such as `svcadm(1M)` or `svcs(1)`, for instructions on which FMRI formats are appropriate.

Service names include prefixes to help identify the purpose of each service. These prefixes include names such as `application`, `device`, `milestone`, `network`, or `system`. The `site` prefix is reserved for site-specific customizations, and services using this prefix will not be delivered in an Oracle Solaris release.

Legacy `init.d` scripts are also represented with FMRIs that start with `lrc` instead of `svc`, for example, `lrc:/etc/rc2_d/S47pppd`. The legacy service's initial start times during system boot are displayed by using the `svcs` command. However, you cannot administer these services by using SMF.

During initial system deployment, services listed in `/etc/inetd.conf` are automatically converted into SMF services. The FMRIs for these services are slightly different. The syntax for a converted `inetd` service is:

`network/`*service-name*`/`*protocol*

In addition, the syntax for a converted service that uses the RPC protocol is:

`network/rpc-`*service-name*`/rpc_`*protocol*

Where *service-name* is the name defined in `/etc/inetd.conf` and *protocol* is the protocol for the service. The `inetconv` command can be used to convert `inetd.conf` entries after initial system deployment.

Service States

The `svcs` command displays the state, start time, and FMRI of service instances. The state of each service is one of the following:

- `degraded` – The service instance is enabled, but is running at a limited capacity.
- `disabled` – The service instance is not enabled and is not running.
- `legacy_run` – The legacy service is not managed by SMF, but the service can be observed. This state is only used by legacy services.
- `maintenance` – The service instance has encountered an error that must be resolved by the administrator.
- `offline` – The service instance is enabled, but the service is not yet running or available to run.
- `online` – The service instance is enabled and has successfully started.
- `uninitialized` – This state is the initial state for all services before their configuration has been read.

An asterisk "*" is appended to the state for instances in transition. A question mark "?" is displayed if the state is absent or unrecognized.

SMF Manifests

An SMF *manifest* is an XML file that describes a service and a set of instances. Manifests are imported to load the properties of that service and its instances into the service configuration repository. See the `service_bundle(4)` man page for a complete description of the contents of an SMF manifest.

The preferred location for manifests is `/lib/svc/manifest`. Manifests stored there will be imported and upgraded by the `svc:/system/early-manifest-import:default` service during the boot process before any services start. Running the import process early ensures that the

repository will contain information from the latest manifests before the services are started. At other times you can import information from these manifests by running this command: `svcadm restart manifest-import`. `/var/svc/manifest` remains available for compatibility purposes, but manifests located there will not be imported or upgraded until the `svc:/system/manifest-import:default` service runs.

Do not make changes to manifests delivered by Oracle or third-party software vendors. Do not directly edit those manifests in `/lib/svc/manifest` and `/var/svc/manifest`, as any customizations will be lost upon upgrade. Instead, either create a site profile to customize the service, or use the `svccfg` or `inetadm` command to manipulate the properties directly. The `/lib/svc/manifest/site` and `/var/svc/manifest/site` directories are also reserved for site-specific use. The Oracle Solaris release will not deliver manifests into those directories.

In the Oracle Solaris 11 release, multiple manifests can be used to describe a single service. This can be useful, for example, to define a new instance of a service without modifying the service's existing manifest. If the same property for the same service or instance is defined by multiple manifests, SMF cannot determine which value to use. When this type of conflict is detected, the instance is placed in the maintenance state.

SMF Profiles

An SMF profile is an XML file that allows customization of services and instances that are delivered by the system. Profiles are available for customization by using a file rather than a set of scripts, or to customize the configuration at deployment or installation time.

All configurations can be customized by using a profile, including adding instances for system-supplied services.

Local customizations must be placed in files named with a `.xml` suffix in the `/etc/svc/profile/site` directory. All customizations in this directory are applied when the system is booted or when the `svcadm restart manifest-import` command is run.

As with manifests, any conflicting definitions between files in `/etc/svc/profile/site` are treated as conflicts, and the affected instances are placed in the maintenance state.

A system profile is also applied during installation. Changes to the system profile in `/etc/svc/profile/generic.xml` are rarely necessary. See the `smf_bootstrap(5)` man page for more information.

For more information about using profiles, see "How to Apply an SMF Profile" on page 129.

Service Configuration Repository

The *service configuration repository* stores persistent configuration information as well as SMF runtime data for services. The repository is distributed among local memory and local files. The service configuration repository can only be manipulated or queried by using SMF interfaces.

For more information about manipulating and accessing the repository, see the `svccfg(1M)` and `svcprop(1)` man pages. The service configuration repository daemon is covered in the `svc.configd(1M)` man page. The service configuration library is documented in the `libscf(3LIB)` man page.

Properties in the repository can be defined on either the service or the instance. Properties that are set on the service are shared by all instances of that service. Properties that are set on the instance are used only by that instance and can override properties on the service.

The `svccfg` command offers a *raw* view of properties, and is precise about whether the properties are set on the service or the instance. If you view a service by using the `svccfg` command, you cannot see instance properties. If you view the instance instead, you cannot see service properties. The `svcprop` command offers a *composed* view of the instance, where both instance properties and service properties are combined into a single property namespace. When service instances are started, the composed view of their properties is used.

All SMF configuration changes can be logged by using the Oracle Solaris auditing framework. Refer to "Configuring the Audit Service (Task Map)" in *Oracle Solaris Administration: Security Services* for more information.

SMF Repository Backups

SMF automatically takes the following backups of the repository:

- The boot backup is taken immediately before the first change to the repository is made during each system startup.
- The `manifest_import` backups occur after `svc:/system/early-manifest-import:default` or `svc:/system/manifest-import:default` completes, if the service imported any new manifests or ran any upgrade scripts.

Four backups of each type are maintained by the system. The system deletes the oldest backup, when necessary. The backups are stored as `/etc/svc/repository-`*type*`-`*YYYYMMDD_HHMMSWS*, where *YYYYMMDD* (year, month, day) and *HHMMSS* (hour, minute, second), are the date and time when the backup was taken. Note that the hour format is based on a 24–hour clock.

You can restore the repository from these backups, if an error occurs. To do so, use the `/lib/svc/bin/restore_repository` command. For more information, see "How to Repair a Corrupt Repository" on page 138.

SMF Snapshots

The data in the service configuration repository includes *snapshots*, as well as a configuration that can be edited. Data about each service instance is stored in the snapshots. The standard snapshots are as follows:

- `initial` – Taken on the first import of the manifest
- `running` – Taken when `svcadm refresh` is run.
- `start` – Taken at the last successful start

The SMF service always executes with the `running` snapshot. This snapshot is automatically created if it does not exist.

The `svccfg` command is used to change current property values. Those values become visible to the service when the `svcadm` command is run to integrate those values into the running snapshot. The `svccfg` command can also be used to, view or revert to instance configurations in another snapshot.

SMF Administrative Layers

In the Oracle Solaris 11 release, information that records the source of properties, property groups, instances, and services has been added to the service configuration repository. This information enables users to determine which data are administrative customizations and which data were delivered with the software.

To help identify the source of an entity, the following layers are defined:

- The `admin` layer includes any changes that are made by using the SMF commands or by calling the `libscf(3LIB)` API.
- The `site-profile` layer includes any values from the files in the `/etc/svc/profile/site` directory or the legacy `/etc/svc/profile/site.xml` and `/var/svc/profile/site.xml` profiles.
- The `system-profile` layer includes any values from the system profile locations: `/etc/svc/profile/generic.xml` and `/etc/svc/profile/platform.xml`.
- The `manifest` layer includes values from a system manifest directory: `/lib/svc/manifest` or `/var/svc/manifest`.

To maintain compatibility for existing clients who expect a single property per property name, as well as to create a policy for overrides, the layering has a simple override behavior. The `admin` layer takes precedence. If a property has a value in the `admin` layer, that is the value that is used by the service. If not, the `site-profile` layer checked, followed by the `system-profile` layer, and finally the `manifest` layer. This behavior allows for local customizations to take precedence over the values that are provided when the system was installed.

These layers are managed automatically by the system. An administrator's direct changes to the repository appear only in the `admin` layer. Other layers are changed only by placing or removing files in standard locations. When a property is placed into the repository due to file contents, the information about that property includes the name of the file that the contents came from.

An administrator cannot modify the lower layers directly by using `svccfg` or `libscf` calls. When the `svccfg delete`, `svccfg delpg`, or `svccfg delprop` command is used, the entity will be masked instead of fully deleted. Normally, users cannot see the deleted entity, but masked entities can be explicitly explored by using the `svccfg listcust` command, and unmasked by using the `svccfg delcust` command, if desired.

The `svccfg listprop` command has options to enable the exploration of these layers. For example, `svccfg listprop -l all` prints all layers and the values in each layer. In addition, the `svccfg listcust` command can be used to list customizations only.

SMF Service Error Logging

Service-specific information, including errors the service or its methods emits, as well as information about enable actions, start times, and so on, are logged in individual files for each service instance in `/var/svc/log`. To determine the name of a service's log file, run the `svcs -x` *service* command.

By default, SMF writes log messages to the `syslog` program and the console only if administrative intervention is required, for example, if a service enters the maintenance state. Other options are available but rarely used. See the `svc.startd(1M)` man page for other potential configurations.

In addition, to error logging, the SMF service can be configured to notify you when an FMA event occurs or when services transition in to or out of a service state. These notifications can use the Simple Network Management Protocol (SNMP) or the Simple Mail Transfer Protocol (SMTP). See "How to Set Up Email Notification of SMF Transition Events" on page 122 for information about setting up SMF notifications.

SMF Administrative and Programming Interfaces

This section introduces the interfaces that are available when you use SMF.

SMF Command-Line Administrative Utilities

SMF provides a set of command-line utilities that interact with SMF and accomplish standard administrative tasks. The following utilities can be used to administer SMF.

TABLE 6–1 Service Management Facility Utilities

Command Name	Function
`inetadm`	Provides the ability to observe or configure services controlled by `inetd`.
`svcadm`	Provides the ability to perform common service management tasks, such as enabling, disabling, or restarting service instances.
`svccfg`	Provides the ability to directly display and manipulate the contents of the service configuration repository. Note: The contents displayed by this command is raw, so it only shows information for the selected entity.
`svcprop`	Retrieves property values from the service configuration repository with an output format appropriate for use in shell scripts. Note: The contents displayed by this command is composed, so it includes information from many entities.
`svcs`	Gives detailed views of the service state of all service instances in the service configuration repository.

Service Management Configuration Library Interfaces

SMF provides a set of programming interfaces, the `libscf(3LIB)` API, that is used to interact with the service configuration repository through the `svc.configd` daemon. This daemon is the arbiter of all requests to the local repository datastores. A set of fundamental interfaces is defined as the lowest level of interaction possible with services in the service configuration repository. The interfaces provide access to all service configuration repository features such as transactions and snapshots.

Many developers only need a set of common tasks to interact with SMF. These tasks are implemented as convenience functions on top of the fundamental services to ease the implementation burden.

SMF Components

SMF includes a master restarter daemon and delegated restarters. In addition, each service or service instance can store configuration data in properties. These properties are organized into property groups to make administration simpler.

SMF Master Restarter Daemon

The `svc.startd` daemon is the master process starter and restarter. The daemon is responsible for managing service dependencies for the entire system. The daemon takes on the previous responsibility that `init` held of starting the appropriate `/etc/rc*.d` scripts at the appropriate

run levels. First, `svc.startd` retrieves the information in the service configuration repository. Next, the daemon starts services when their dependencies are met. The daemon is also responsible for restarting services that have failed and for shutting down services whose dependencies are no longer satisfied. The daemon uses operating system events, such as process death, to keep track of service states.

SMF Delegated Restarters

Some services have a set of common behaviors on startup. To provide commonality among these services, a delegated restarter might take responsibility for these services. In addition, a delegated restarter can be used to provide more complex or application-specific restarting behavior. The delegated restarter can support a different set of methods, but exports the same service states as the master restarter. The restarter's name is stored with the service. A current example of a delegated restarter is `inetd`, which can start Internet services on demand, rather than having the services always running. The defined restarter for each service can be displayed using the `svcs -l` command.

SMF Properties and Property Groups

All information in the service configuration repository is stored as a set of properties, which are grouped by property groups. *Property groups* are used to group different types of service information. Some of the common property groups include:

- `general` – Contains information about the service instance, including the `general/enabled` property, which defines whether the instance is enabled
- `restarter` – Contains runtime information that is stored by the service's restarter, including the `restarter/state` property, which shows the current state of the service
- `start` – Contains the start method definition, including the `start/exec` property, which defines what program to execute to start the service

The system defines many other property groups. Services can also define their own property groups in order to store service-specific configuration information. Another common property group is `config`, although others are common as well. See the `smf(5)` man page for more information about properties and property groups.

Managing Information in the Service Configuration Repository

There are many ways to view, modify, or delete information by using SMF. This section discusses which methods are best for which uses.

Viewing SMF Information

You can use the `svccfg` and `svcprop` commands to view information in the service configuration repository. For a complete description of these commands, see the `svccfg(1M)` and `svcprop(1)` man pages.

- `svcprop` – Lists the values assigned to property groups or properties in running snapshot. Because this command combines data for the service, and service instance, it provides a comprehensive view of the data.
- `svccfg listpg` – Lists information about property groups in the selected service or service instance. All property group names, types, and flags are listed.
- `svccfg listprop` – Lists information about properties and property group in the selected service or service instance. For property groups, the names, types and flags are listed. For properties, the names, types and values are listed.
  - `-l layer_name` – Lists the properties and property groups within the named layer in a service or service instance. Using `all` as a layer name lists all of the layers and properties for that service. Note that each service and service instance is displayed separately.
  - `-f` – Lists the file name that a property came from.
  - `-o` – Selects the fields to display.
- `svccfg listcust` – Lists any site customizations in the `site-profile` or `admin` layer. Also lists any masked entries for the selected service or service instance.
  - `-M` – Lists only masked entities.
  - `-L` – Shows all local customizations, which includes both administrative customizations and site profile customizations.

Modifying SMF Information

You can modify information about a service or service instance by using a manifest, or a profile, or by using the `svccfg` command. Any changes you make with the `svccfg` command are recorded in the `admin` layer. The following list includes some of the options that you can use to modify information by using the `svccfg` command:

- `addpg` – Adds a property group to the selected service or service instance
- `addpropvalue` – Assigns an additional value to a existing property

- `setenv` – Sets an environment variable for a service or service instance
- `setprop` – Sets the value of a named property in the selected service
- `setnotify` – Sets notification parameters for software events and FMA event classes

Deleting SMF Information

You can delete information about a service or service instance by using the `svccfg` command. The following list includes some of the options that you can use to remove information by using the `svccfg` command:

- `delcust` – Deletes any administrative customizations for the selected service
- `delpropvalue` – Deletes all property values that match the given string
- `unsetenv` – Removes an environment variable for a service or service instance

When you delete information from the repository one of two things happen, either the entity will be hidden or the entity will be removed. Any information that has been defined only in the `admin` layer will be removed. Any information from a manifest or profile will be hidden, so that the standard commands will not display the information. The information is hidden so that if you need to undo the deletion, you readily will have the information available.

SMF and Booting

When a system is being booted, you can select the milestone to boot to or the level of error messages to be recorded as follows:

- You can choose a specific milestone to boot to by using this command:

  ```
  ok boot -m milestone=milestone
  ```

  The default is `all`, which starts all enabled services. Also useful is `none`, which starts only `init`, `svc.startd`, and `svc.configd`, and provides a debugging environment where services can be started manually. See "How to Boot Without Starting Any Services" on page 141 for instructions on how to use the `none` milestone.

  The run level equivalents `single-user`, `multi-user`, and `multi-user-server` are also available, but are not commonly used. In particular, `multi-user-server`does not start any services that are not dependent on that milestone, so might not include important services.

- You can choose the level of logging for `svc.startd` by using this command:

  ```
  ok boot -m logging-level
  ```

  The logging levels that you can select include `quiet` and `verbose`. See "SMF Service Error Logging" on page 110 for specific information about the logging levels.

SMF Compatibility

Although many standard services are now managed by SMF, the scripts placed in `/etc/rc*.d` continue to be executed on run level transitions. Most of the `/etc/rc*.d` scripts that were included in previous releases have been removed as part of SMF. The ability to continue to run the remaining scripts allows for third-party applications to be added without having to convert the services to use SMF.

In addition, `/etc/inittab` entries also continue to be processed by the `init` command. Also, `/etc/inetd.conf` is available for packages to amend. During initial system deployment, services that are listed in `/etc/inetd.conf` are automatically converted into SMF services. Any later additions can be converted by using the `inetconv` command. The status of these services can be viewed, but no other changes are supported through SMF. Applications that use this conversion feature will not benefit from the precise fault containment provided by SMF. The latest version of `inetd` does not look for entries in `/etc/inetd.conf` to convert after the initial boot.

Applications that are converted to utilize SMF no longer need to make use of the mechanisms listed in this section.

Run Levels

A system's *run level* (also known as an *init state*) defines what services and resources are available to users. A system can be in only one run level at a time.

The release has eight run levels, which are described in the following table. The default run level is specified in the `/etc/inittab` file as run level 3.

TABLE 6–2 Oracle Solaris Run Levels

Run Level	Init State	Type	Purpose
0	Power-down state	Power-down	To shut down the operating system so that it is safe to turn off power to the system.
s or S	Single-user state	Single-user	To run as a single user with some file systems mounted and accessible.
1	Administrative state	Single-user	To access all available file systems. User logins are disabled.
2	Multiuser state	Multiuser	For normal operations. Multiple users can access the system and all file system. All daemons are running except for the NFS and SMB server daemons.
3	Multiuser level with NFS resources shared	Multiuser	For normal operations with NFS and SMB resources shared. This is the default run level.

TABLE 6–2 Oracle Solaris Run Levels *(Continued)*

Run Level	Init State	Type	Purpose
4	Alternative multiuser state	Multiuser	Not configured by default, but available for customer use.
5	Power-down state	Power-down	To shut down the operating system so that it is safe to turn off power to the system. If possible, automatically turns off power on systems that support this feature.
6	Reboot state	Reboot	To shut down the system to run level 0, and then reboot to multiuser level with NFS and SMB resources shared (or whatever level is the default in the `inittab` file).

In addition, the `svcadm` command can be used to change the run level of a system, by selecting a milestone at which to run. The following table shows which run level corresponds to each milestone.

TABLE 6–3 Run Levels and SMF Milestones

Run Level	SMF Milestone FMRI
S	`milestone/single-user:default`
2	`milestone/multi-user:default`
3	`milestone/multi-user-server:default`

When to Use Run Levels or Milestones

In general, changing milestones or run levels is an uncommon procedure. If it is necessary, using the `init` command to change to a run level will change the milestone as well and is the appropriate command to use. The `init` command is also useful for shutting down a system.

However, booting a system by using the `none` milestone can be very useful when you are debugging startup problems. There is no equivalent run level to the `none` milestone. See "How to Boot Without Starting Any Services" on page 141 for specific instructions.

Determining a System's Run Level

Display run level information by using the `who -r` command.

```
$ who -r
```

Use the `who -r` command to determine a system's current run level for any level.

EXAMPLE 6–1 Determining a System's Run Level

This example displays information about a system's current run level and previous run levels.

```
$ who -r
 .    run-level 3  Dec 13 10:10  3  0 S
$
```

Output of `who -r` command	Description
`run-level 3`	Identifies the current run level
`Dec 13 10:10`	Identifies the date of last run level change
`3`	Also identifies the current run level
`0`	Identifies the number of times the system has been at this run level since the last reboot
`S`	Identifies the previous run level

/etc/inittab File

When you boot the system or change run levels with the `init` or `shutdown` command, the `init` daemon starts processes by reading information from the `/etc/inittab` file. This file defines these important items for the `init` process:

- That the `init` process will restart
- What processes to start, monitor, and restart if they terminate
- What actions to take when the system enters a new run level

Each entry in the `/etc/inittab` file has the following fields:

id:*rstate*:*action*:*process*

The following table describes the fields in an `inittab` entry.

TABLE 6–4 Fields Descriptions for the `inittab` File

Field	Description
id	Is a unique identifier for the entry.
rstate	Lists the run levels to which this entry applies.
action	Identifies how the process that is specified in the process field is to be run. Possible values include: `sysinit`, `boot`, `bootwait`, `wait`, and `respawn`. For a description of the other action keywords, see `inittab(4)`.
process	Defines the command or script to execute.

EXAMPLE 6–2 Default `inittab` File

The following example shows a default `inittab` file that is installed with the release. A description for each line of output in this example follows.

```
ap::sysinit:/usr/sbin/autopush -f /etc/iu.ap      (1)
smf::sysinit:/lib/svc/bin/svc.startd    >/dev/msglog 2<>/dev/msglog </dev/console          (2)
p3:s1234:powerfail:/usr/sbin/shutdown -y -i5 -g0 >/dev/msglog 2<>/dev/...    (3)
```

1. Initializes STREAMS modules
2. Initializes the master restarter for SMF
3. Describes a power fail shutdown

What Happens When the System Is Brought to Run Level 3

1. The `init` process is started and reads the properties that are defined in the `svc:/system/environment:init` SMF service to set any environment variables.
2. The `init` process reads the `inittab` file and does the following:
   a. Executes any process entries that have `sysinit` in the `action` field so that any special initializations can take place before users login
   b. Passes the startup activities to `svc.startd`

   For a detailed description of how the `init` process uses the `inittab` file, see the `init(1M)` man page.

◆ ◆ ◆ CHAPTER 7

Managing Services (Tasks)

This chapter covers the tasks required to manage and monitor the Service Management Facility (SMF). In addition, information about managing run level scripts is provided. The following topics are covered:

- "Monitoring SMF Services" on page 120
- "Managing SMF Services" on page 125
- "Configuring SMF Services" on page 129
- "Using Run Control Scripts" on page 135
- "Troubleshooting the Service Management Facility" on page 138

Monitoring Services (Task Map)

The following task map describes the procedures that are needed to monitor SMF services.

Task	Description	For Instructions
Display the status of a service instance.	Displays the status of all running service instances.	"How to List the Status of a Service" on page 120
Display the customizations to a service.	Displays the customizations in the `admin` layer for the service.	"How to List Customizations of a Service" on page 121
Display the service dependents.	Display the services that are dependent on the specified service.	"How to Show Which Services Are Dependent on a Service Instance" on page 121
Display the dependencies of a service.	Display the services that a specified service is dependent on. This information can be used to help identify what is preventing a service from starting.	"How to Show Which Services a Service Is Dependent On" on page 122

Task	Description	For Instructions
Configure email notification for significant events.	Uses SNMP or SMTP to send a notification of a change in service state or of an FMA event.	"How to Set Up Email Notification of SMF Transition Events" on page 122

Monitoring SMF Services

The following tasks show how to monitor SMF services.

▼ How to List the Status of a Service

This procedure can be used to show what services are running.

- **Run the `svcs` command.**

  Running this command without any options displays a status report of the service specified by the FMRI.

  ```
  $ svcs -l FMRI
  ```

Example 7–1 Showing the Status of the `sendmail` Service

This example shows the status of a service that includes dependencies. Also, the `-p` option is used so that information about the process ID, start time and command run is displayed

```
$ svcs -lp network/smtp:sendmail
fmri         svc:/network/smtp:sendmail
name         sendmail SMTP mail transfer agent
enabled      true
state        online
next_state   none
state_time   Tue Aug 09 19:25:54 2011
logfile      /var/svc/log/network-smtp:sendmail.log
restarter    svc:/system/svc/restarter:default
contract_id  114
manifest     /etc/svc/profile/generic_limited_net.xml
manifest     /lib/svc/manifest/network/smtp-sendmail.xml
dependency   require_all/refresh file://localhost/etc/mail/sendmail.cf (online)
dependency   require_all/refresh file://localhost/etc/nsswitch.conf (online)
dependency   optional_all/none svc:/system/filesystem/autofs (online)
dependency   require_all/none svc:/system/filesystem/local (online)
dependency   require_all/none svc:/network/service (online)
dependency   require_all/refresh svc:/milestone/name-services (online)
dependency   optional_all/refresh svc:/system/identity:domain (online)
dependency   optional_all/none svc:/system/system-log (online)
process      101077 /usr/lib/sendmail -bd -q15m
```

Example 7–2 Showing the Status of all Services

The following command lists all services that are installed on the system as well as the status of each service. The command displays those services that are disabled as well as those that are enabled.

```
$ svcs -a
```

Example 7–3 Showing the Status of Services Controlled by `inetd`

The following command lists services that are controlled by `inetd`. Each service's FMRI is listed, along with the run state and whether the service is enabled or disabled.

```
$ inetadm
```

▼ How to List Customizations of a Service

- **List local customizations.**

  This command displays all of the changes at the `admin` layer for the selected service.

  ```
  % /usr/sbin/svccfg -s FMRI listcust
  ```

▼ How to Show Which Services Are Dependent on a Service Instance

This procedure shows how to determine which service instances depend on the specified service.

- **Display the service dependents.**

  ```
  $ svcs -D FMRI
  ```

Example 7–4 Displaying the Service Instances That Are Dependent on the `multiuser` Milestone

The following example shows how to determine which service instances are dependent on the `multiuser` milestone.

```
$ svcs -D milestone/multi-user
STATE          STIME    FMRI
disabled       Aug_09   svc:/application/time-slider:default
disabled       Aug_09   svc:/application/management/net-snmp:default
online         Aug_09   svc:/system/intrd:default
online         Aug_09   svc:/system/boot-config:default
online         Aug_09   svc:/milestone/multi-user-server:default
```

How to Show Which Services a Service Is Dependent On

This procedure shows how to determine which services a specified service instance is dependent on.

- **Display the service dependencies.**

  ```
  $ svcs -d FMRI
  ```

Example 7–5 Displaying the Service Instances That the `multiuser` Milestone Is Dependent On

The following example shows the services instances that the `multiuser` milestone is dependent on:

```
$ svcs -d milestone/multi-user:default
STATE          STIME    FMRI
disabled       Aug_09   svc:/network/nfs/client:default
disabled       Aug_09   svc:/network/smb/client:default
disabled       Aug_09   svc:/system/mdmonitor:default
disabled       Aug_09   svc:/application/print/server:default
online         Aug_09   svc:/system/resource-controls:default
online         Aug_09   svc:/system/metasync:default
online         Aug_09   svc:/system/rmtmpfiles:default
online         Aug_09   svc:/system/utmp:default
online         Aug_09   svc:/system/name-service/cache:default
online         Aug_09   svc:/system/device/audio:default
online         Aug_09   svc:/network/rpc/bind:default
online         Aug_09   svc:/milestone/name-services:default
online         Aug_09   svc:/network/iscsi/initiator:default
online         Aug_09   svc:/milestone/single-user:default
online         Aug_09   svc:/system/filesystem/local:default
online         Aug_09   svc:/network/inetd:default
online         Aug_09   svc:/system/cron:default
online         Aug_09   svc:/system/filesystem/autofs:default
online         Aug_09   svc:/system/filesystem/ufs/quota:default
online         Aug_09   svc:/system/power:default
online         Aug_09   svc:/system/system-log:default
online         Aug_09   svc:/system/system-log:default
online         Aug_09   svc:/system/auditd:default
online         Aug_09   svc:/network/smtp:sendmail
```

How to Set Up Email Notification of SMF Transition Events

This procedure causes the system to generate an email notification each time one of the services or a selected service has a change in state. You can choose to use either SMTP or SNMP. Normally, you would only select SNMP if you already have SNMP configured for some other reason.

By default, SNMP traps are sent on maintenance transitions. If you use SNMP for monitoring, you can configure additional traps for other state transitions.

1 **Become an administrator or assume a role that includes the `Service Management` rights profile.**

For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2 **Set notification parameters.**

The following examples show how to set notification parameters for SMF and FMA events, as well as how to list and delete notification parameters.

Example 7–6 Configuring Notifications for All SMF Service State Events

The following command creates a notification that sends email when transactions go into the `maintenance` state.

```
# /usr/sbin/svccfg setnotify -g maintenance mailto:sysadmins@example.com
```

You can also choose to select transactions that start in the state listed with the `from` option and end in the state listed with the `to` option. The valid SMF states for this option are: `degraded`, `maintenance`, `offline`, and `online`. You can use the `-g all` option to generate email for all state transition events. See the Notification Parameters section in the `smf(5)` man page for more information.

Example 7–7 Configuring Notifications for an Individual Service

The following command creates a notification that sends email when the `switch` service goes into the `online` state.

```
# /usr/sbin/svccfg -s svc:/system/name-service/switch:default setnotify to-online \
 mailto:sysadmins@example.com
```

Example 7–8 Configuring Notifications for FMA Events

The following command creates a notification that sends an SNMP message when a FMA problem is repaired.

```
# /usr/sbin/svccfg setnotify problem-repaired snmp:
```

The FMA event classes include `problem-diagnosed`, `problem-updated`, `problem-repaired` and `problem-resolved`. See the Notification Parameters section in the `smf(5)` man page for more information.

Example 7–9 Listing Notification Settings

The following command shows the notification settings for a new problem diagnosed by the FMA service. Notification settings for SMF service state transition events can be displayed by including the service state instead of the event class or by not including any arguments with `listnotify`.

```
# /usr/sbin/svccfg listnotify problem-diagnosed
```

Example 7–10 Deleting Notification Settings

The following command deletes the notification settings associated with the `switch` service transitioning to the `online` service state. You can use an FMA event class in place of the service state.

```
# /usr/sbin/svccfg -s svc:/system/name-service/switch:default delnotify to-online
```

Managing SMF Services (Task Map)

The following task map describes the procedures that are needed to manage SMF services.

Task	Description	For Instructions
Disable a service instance.	Stops a running service and prevents the service from restarting.	"How to Disable a Service Instance" on page 125
Enable a service instance.	Starts a service. In addition, the service will be restarted during subsequent reboots.	"How to Enable a Service Instance" on page 125
Restart a service.	Restarts a service without having to use separate commands to disable and then enable the service.	"How to Restart a Service" on page 126
Restore a service in the maintenance state.	Shows how to clean up and restart a service that is in the maintenance state.	"How to Restore a Service That Is in the Maintenance State" on page 127
Create a profile.	Create a profile to easily deploy customizations without running commands.	"How to Create an SMF Profile" on page 127
Apply a profile.	Uses the information in a profile to disable, enable, or customize services as needed.	"How to Apply an SMF Profile" on page 129

Managing SMF Services

This section includes information on managing SMF services.

Using RBAC Rights Profiles With SMF

You can use RBAC rights profiles to allow users to manage some of the SMF services, without having to give the user root access. The rights profiles define what commands the user can run. For SMF, the following profiles have been created:

- `Service Management`: User can add, delete or modify services.
- `Service Operator`: User can request state changes of any service instance, such as restart and refresh.

For specific information about the authorizations, see the `smf_security(5)` man page. For instructions to assign a rights profile, see "How to Change the RBAC Properties of a User" in *Oracle Solaris Administration: Security Services*.

▼ How to Disable a Service Instance

Use the following procedure to disable a service. The service status change is recorded in the service configuration repository. Once the service is disabled, the disabled state will persist across reboots. The only way to get the service running again is to enable it.

1. **Become an administrator or assume a role that includes the `Service Management` rights profile.**

   For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2. **Check the dependents of the service you want to disable.**

   If this service has dependents that you need, then you cannot disable this service.

   ```
   # svcs -D FMRI
   ```

3. **Disable the service.**

   ```
   # svcadm disable FMRI
   ```

▼ How to Enable a Service Instance

Use the following procedure to enable a service. The service status change is recorded in the service configuration repository. After the service is enabled, the enabled state will persist across system reboots, however the service will start only if all dependencies are met.

1. **Become an administrator or assume a role that includes the `Service Management` rights profile.**

   For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2. **Enable a service.**

   ```
   # svcadm enable FMRI
   ```

3. **Verify that the service has been enabled.**

   ```
   # svcs -x FMRI
   ```

4. **Enable service dependencies.**

   If required dependencies are disabled, enable them with the following command:

   ```
   # svcadm enable -r FMRI
   ```

Example 7–11 Enabling a Service in Single-user Mode

The following command enables `rpcbind`. The `-t` option starts the service in temporary mode which does not change the service repository, so this change will not persist across a reboot. The `-r` option recursively starts all the dependencies of the named service.

```
# svcadm enable -rt rpc/bind
```

▼ How to Restart a Service

If a service is currently running but needs to be restarted due to a configuration change or some other reason, the service can be restarted without you having to type separate commands to stop and start the service. The only reason to specifically disable and then enable a service is if changes need to be made before the service is enabled, and after the service is disabled.

1. **Become an administrator or assume a role that includes the `Service Management` rights profile.**

   For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2. **Restart a service.**

   ```
   # svcadm restart FMRI
   ```

How to Restore a Service That Is in the Maintenance State

1 **Become an administrator or assume a role that includes the `Service Management` rights profile.**

For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2 **Determine why the service is in maintenance.**

```
# svcs -x FMRI
```

Consult the log file or man page mentioned to determine what the error is.

3 **Determine if any process that are dependent to the service have not stopped.**

Normally, when a service instance is in a maintenance state, all processes associated with that instance have stopped. However, you should make sure before you proceed. The following command lists all of the processes that are associated with a service instance as well as the PIDs for those processes.

```
# svcs -p FMRI
```

4 **(Optional) Kill any remaining processes.**

Repeat this step for all processes that are displayed by the `svcs` command.

```
# pkill -9 process-name
```

5 **Restore the service.**

```
# svcadm clear FMRI
```

How to Create an SMF Profile

A profile is an XML file which lists SMF services and whether each should be enabled or disabled. Profiles are used to enable or disable many services at once. Profiles are also used to set property values, add property values, and even create a service and instances of a service. Not all services need to be listed in a profile. Each profile only needs to include those services that need to be enabled or disabled to make the profile useful.

1 **Become an administrator or assume a role that includes the `Service Management` rights profile.**

For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2 Create a profile.

In this example, the `svccfg` command is used to create a profile that represents the local customizations on the current system. Alternatively, you could make a copy of an existing profile to edit.

```
# svccfg extract> profile.xml
```

If you have large numbers of identical systems, or if you want to archive the system configuration for later restoration, you might want to use this procedure to create a unique version of a SMF profile.

3 Edit the `profile.xml` file to make any required changes.

a. Change the name of the profile in the `service_bundle` declaration.

In this example the name is changed to `profile`.

```
# cat profile.xml
  ...
<service_bundle type='profile' name='profile'
    xmIns::xi='http://www.w3.org/2003/XInclude'
  ...
```

b. Remove any services that should not be managed by this profile.

For each service, remove the three lines that describe the service. Each service description starts with `<service` and ends with `</service`. This example shows the lines for the LDAP client service.

```
# cat profile.xml
 ...
 <service name='network/ldap/client' version='1' type='service'>
        <instance  name='default' enabled='true'/>
 </service>
```

c. Add any services that should be managed by this profile.

Each service needs to be defined using the three line syntax shown above.

d. If necessary, change the enabled flag for selected services.

In this example, the `sendmail` service is disabled.

```
# cat profile.xml
  ...
  <service  name='network/smtp' version='1' type='service'>
    <instance  name='sendmail' enabled='false'/>
  </service>
  ...
```

4 When necessary, apply the new profile.

See "How to Apply an SMF Profile" on page 129 for instructions.

▼ How to Apply an SMF Profile

1. **Become an administrator or assume a role that includes the `Service Management` rights profile.**

   For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2. **Apply a profile.**

   In this example, the `profile.xml` profile is applied.

   ```
   # svccfg apply profile.xml
   ```

Configuring SMF Services (Task Map)

The following task map describes the procedures that are needed to configure SMF services.

Task	Description	For Instructions
Modify a service.	Modifies a service property of a specified service instance.	"How to Modify an SMF Service Property" on page 130
	Modifies the configuration file of a service instance.	"How to Modify a Service That Is Configured by a File" on page 130
	Modifies an environment variable of a service instance.	"How to Change an Environment Variable for a Service" on page 131
	Deletes the customizations in the `admin` layer for a service.	"How to Delete Customizations for a Service" on page 133
Modify an `inetd` service.	Changes a configuration property of a service controlled by `inetd`.	"How to Change a Property for an `inetd` Controlled Service" on page 131
	Changes the startup options of a service controlled by `inetd`.	"How to Modify a Command-Line Argument for an `inetd` Controlled Service" on page 133
Convert `inetd.conf` entries.	Converts `inetd` services into legacy-run services that can be monitored using SMF.	"How to Convert `inetd.conf` Entries" on page 134

Configuring SMF Services

The following tasks show how to configure SMF services. In particular, how to modify service properties and other configuration information for a service or a service instance.

How to Modify an SMF Service Property

This procedure shows how to modify the property that identifies the user that can start a service.

1. **Become an administrator or assume a role that includes the `Service Management` rights profile.**

   For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2. **Change the value that is assigned to the `start/user` property.**

   First, give the FMRI to identify the appropriate service. Next, assign the UID that will start the service.

   ```
   # svccfg -s FMRI
   svc:/service: setprop start/user = astring: newlogin
   ```

3. **Refresh the service.**

   ```
   # svcadm refresh FMRI
   ```

How to Modify a Service That Is Configured by a File

The following procedure shows how to change the configuration of a service that is not managed by the `inetd` service.

1. **Become an administrator or assume a role that includes the `Service Management` rights profile.**

   For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2. **Make changes to the configuration files, as needed.**

   Many of the services have one or more configuration files that are used to define the startup or other configuration information. These files can be changed while the service is running. The contents of the files is only checked when the service is started.

3. **Restart the service.**

   ```
   # svcadm restart FMRI
   ```

Example 7–12 Adding a New NTP Server

To add a new NTP server to support your NTP clients, add a new entry for the server to the `/etc/inet/ntp.conf` file. Next, restart the NTP service. This example shows you what the `ntp.conf` file could look like, as well as how to restart the service.

```
# cat /etc/inet/ntp.conf
   .
   .
```

```
server ntpserver1.example.com
server ntpserver2.example.com
# svcadm restart svc:/network/ntp:default
```

How to Change an Environment Variable for a Service

This procedure shows how to modify `cron` environment variables to help with debugging.

1. **Become an administrator or assume a role that includes the `Service Management` rights profile.**

   For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2. **Verify that the service is running.**

   ```
   # svcs system/cron
   STATE          STIME    FMRI
   online         Dec_04   svc:/system/cron:default
   ```

3. **Set environment variables.**

   In this example the `UMEM_DEBUG` and `LD_PRELOAD` environment variables are set. For information about the `setenv` subcommand refer to the `svccfg(1M)` man page.

   ```
   # svccfg -s system/cron:default setenv UMEM_DEBUG default
   # svccfg -s system/cron:default setenv LD_PRELOAD libumem.so
   ```

4. **Refresh and restart the service.**

   ```
   # svcadm refresh system/cron
   # svcadm restart system/cron
   ```

5. **Verify that the change has been made.**

   ```
   # pargs -e ‘pgrep -f /usr/sbin/cron‘
   100657: /usr/sbin/cron
   envp[0]: LOGNAME=root
   envp[1]: LD_PRELOAD=libumem.so
   envp[2]: PATH=/usr/sbin:/usr/bin
   envp[3]: SMF_FMRI=svc:/system/cron:default
   envp[4]: SMF_METHOD=/lib/svc/method/svc-cron
   envp[5]: SMF_RESTARTER=svc:/system/svc/restarter:default
   envp[6]: TZ=GB
   envp[7]: UMEM_DEBUG=default
   ```

How to Change a Property for an `inetd` Controlled Service

1. **Become an administrator or assume a role that includes the `Service Management` rights profile.**

   For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2 List the properties for the specific service.

This command displays all of the properties for the service identified by the FMRI.

```
# inetadm -l FMRI
```

3 Change the property for the service.

Each property for an `inetd` controlled service is defined by a property name and an assigned value. Supplying the property name without a specified value resets the property to the default value. Specific information about the properties for a service should be covered in the man page associated with the service.

```
# inetadm -m FMRI property-name=value
```

4 Verify that the property has changed.

List the properties again to make sure that the appropriate change has occurred.

```
# inetadm -l FMRI
```

5 Confirm that the change has taken effect.

Confirm the property change that the change has the desired effect.

Example 7–13 Changing the `tcp_trace` Property for `telnet`

The following example shows how to set the `tcp_trace` property for `telnet` to `true`. Checking the `syslog` output after running a `telnet` command shows that the change has taken effect.

```
# inetadm -l svc:/network/telnet:default
SCOPE    NAME=VALUE
         name="telnet"
 .
 .
default  inherit_env=TRUE
default  tcp_trace=FALSE
default  tcp_wrappers=FALSE
# inetadm -m svc:/network/telnet:default tcp_trace=TRUE
# inetadm -l svc:/network/telnet:default
SCOPE    NAME=VALUE
         name="telnet"
 .
 .
default  inherit_env=TRUE
         tcp_trace=TRUE
default  tcp_wrappers=FALSE
# telnet localhost
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
login: root
Password:
.
.
.
Last login: Mon Jun 21 05:55:45 on console
```

```
Sun Microsystems Inc.   SunOS 5.10      s10_57  May 2004
# ^D
Connection to localhost closed by foreign host.
# tail -1 /var/adm/messages
Jun 21 06:04:57 yellow-19 inetd[100308]: [ID 317013 daemon.notice] telnet[100625]
    from 127.0.0.1 32802
```

How to Delete Customizations for a Service

- **Delete local customizations.**

  This command deletes all of the changes at the `admin` layer for the selected service.

  ```
  % /usr/sbin/svccfg -s FMRI delcust
  ```

How to Modify a Command-Line Argument for an `inetd` Controlled Service

1. **Become an administrator or assume a role that includes the `Service Management` rights profile.**

   For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2. **List the exec property for the specific service.**

   This command displays all the properties for the service identified by the FMRI. Adding the `grep` command restricts the output to the `exec` property for the service.

   ```
   # inetadm -l FMRI|grep exec
   ```

3. **Change the exec property for the service.**

   The *command-syntax* set with the `exec` property defines the command string that is run when the service is started.

   ```
   # inetadm -m FMRI exec="command-syntax"
   ```

4. **Verify that the property has changed.**

   List the properties again to make sure that the appropriate change has occurred.

   ```
   # inetadm -l FMRI
   ```

Example 7–14 Adding the Connection Logging (`-l`) Option to the `ftp` Command

In this example, the `-l` option is added to the `ftp` daemon when it is started. The effect of this change can be seen by reviewing the `syslog` output after a `ftp` login session has been completed.

```
# inetadm -l svc:/network/ftp:default | grep exec
        exec="/usr/sbin/in.ftpd -a"
# inetadm -m svc:/network/ftp:default exec="/usr/sbin/in.ftpd -a -l"
# inetadm -l svc:/network/ftp:default
SCOPE    NAME=VALUE
         name="ftp"
         endpoint_type="stream"
         proto="tcp6"
         isrpc=FALSE
         wait=FALSE
         exec="/usr/sbin/in.ftpd -a -l"
 .
 .
# ftp localhost
Connected to localhost.
220 yellow-19 FTP server ready.
Name (localhost:root): mylogin
331 Password required for mylogin.
Password:
230 User mylogin logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> quit
221-You have transferred 0 bytes in 0 files.
221-Total traffic for this session was 236 bytes in 0 transfers.
221-Thank you for using the FTP service on yellow-19.
221 Goodbye.
# tail -2 /var/adm/messages
Jun 21 06:54:33 yellow-19 ftpd[100773]: [ID 124999 daemon.info] FTP LOGIN FROM localhost
     [127.0.0.1], mylogin
Jun 21 06:54:38 yellow-19 ftpd[100773]: [ID 528697 daemon.info] FTP session closed
```

How to Convert `inetd.conf` Entries

The following procedure converts `inetd.conf` entries into SMF service manifests. This procedure needs to be run any time a third-party application that depends on `inetd` is added to a system. Also run this procedure, if you need to make configuration changes to the entry in `/etc/inetd.conf`.

1. **Become an administrator or assume a role that includes the `Service Management` rights profile.**

   For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2. **Convert the `inetd.conf` entries.**

   The `inetconv` command converts each entry in the selected file into service manifests.

   `#` `inetconv -i` *filename*

Example 7–15 Converting `/etc/inet/inetd.conf` Entries into SMF Service Manifests

```
# inetconv -i /etc/inet/inetd.conf
```

Using Run Control Scripts (Task Map)

The following task map includes several procedures that are associated with using run control scripts. Each row includes a task, a description of when you would want to perform that task, followed by a link to the task.

Task	Description	For Instructions
Stop or start a legacy service.	Use a run control script to stop or start a service.	"How to Use a Run Control Script to Stop or Start a Legacy Service" on page 135
Add a run control script.	Create a run control script and add it to the `/etc/init.d` directory.	"How to Add a Run Control Script" on page 136
Disable a run control script.	Disable a run control script by renaming the file.	"How to Disable a Run Control Script" on page 137

Using Run Control Scripts

The following procedures show how to use run control scripts to stop or start a legacy service. Also included are instructions for adding or removing a run control script.

▼ How to Use a Run Control Script to Stop or Start a Legacy Service

1. **Become an administrator or assume a role that includes the `Service Management` rights profile.**

   For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2. **Stop the system service.**

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

3. **Restart the system service.**

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

4. **Verify that the service has been stopped or started.**

   ```
   # pgrep -f service
   ```

Example 7–16 Using a Run Control Script to Stop or Start a Service

For example, you can stop the NFS server daemons by typing the following:

```
# /etc/init.d/nfs.server stop
# pgrep -f nfs
```

Then, you can restart the NFS server daemons by typing the following:

```
# /etc/init.d/nfs.server start
# pgrep -f nfs
101773
101750
102053
101748
101793
102114
# pgrep -f nfs -d, | xargs ps -fp
     UID    PID   PPID   C    STIME TTY         TIME CMD
  daemon 101748      1   0   Sep 01 ?           0:06 /usr/lib/nfs/nfsmapid
  daemon 101750      1   0   Sep 01 ?          26:27 /usr/lib/nfs/lockd
  daemon 101773      1   0   Sep 01 ?           5:27 /usr/lib/nfs/statd
    root 101793      1   0   Sep 01 ?          19:42 /usr/lib/nfs/mountd
  daemon 102053      1   0   Sep 01 ?        2270:37 /usr/lib/nfs/nfsd
  daemon 102114      1   0   Sep 01 ?           0:35 /usr/lib/nfs/nfs4cbd
```

How to Add a Run Control Script

If you want to add a run control script to start and stop a service, copy the script into the `/etc/init.d` directory. Then, create links in the `rc`*n* `.d` directory where you want the service to start and stop.

See the `README` file in each `/etc/rc` *n*`.d` directory for more information on naming run control scripts. The following procedure describes how to add a run control script.

1. **Become an administrator or assume a role that includes the `Service Management` rights profile.**

   For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2. **Add the script to the `/etc/init.d` directory.**

   ```
   # cp filename/etc/init.d
   # chmod 0744 /etc/init.d/filename
   # chown root:sys /etc/init.d/filename
   ```

3. **Create links to the appropriate `rc` *n*`.d` directory.**

   ```
   # cd /etc/init.d
   # ln filename /etc/rc2.d/Snnfilename
   # ln filename /etc/rcn.d/Knnfilename
   ```

4. **Verify that the script has links in the specified directories.**

   ```
   # ls /etc/init.d/*filename /etc/rc2.d/*filename /etc/rcn.d/*filename
   ```

Example 7–17 Adding a Run Control Script

The following example shows how to add a run control script for the `xyz` service.

```
# cp xyz /etc/init.d
# chmod 0744 /etc/init.d/xyz
# chown root:sys /etc/init.d/xyz
# cd /etc/init.d
# ln xyz /etc/rc2.d/S99xyz
# ln xyz /etc/rc0.d/K99xyz
# ls /etc/init.d/*xyz /etc/rc2.d/*xyz /etc/rc0.d/*xyz
```

▼ How to Disable a Run Control Script

You can disable a run control script by renaming it with an underscore (_) at the beginning of the file name. Files that begin with an underscore or dot are not executed. If you copy a file by adding a suffix to it, both files will be run.

1. **Become an administrator or assume a role that includes the `Service Management` rights profile.**

   For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2. **Rename the script by adding an underscore (_) to the beginning of the new file.**

   ```
   # cd /etc/rcn.d
   # mv filename_filename
   ```

3. **Verify that the script has been renamed.**

   ```
   # ls _*
   _filename
   ```

Example 7–18 Disabling a Run Control Script

The following example shows how to rename the `S99datainit` script.

```
# cd /etc/rc2.d
# mv S99datainit _S99datainit
# ls _*
_S99datainit
```

Troubleshooting the Service Management Facility

The following procedures show how to troubleshoot or fix SMF services.

▼ Debugging a Service That Is Not Starting

In this procedure, the print service is disabled.

1 Become an administrator or assume a role that includes the `Service Management` rights profile.

For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2 Request information about the service that is not running due to an error.

```
# svcs -xv
svc:/application/print/server:default (LP Print Service)
 State: disabled since Wed 13 Oct 2004 02:20:37 PM PDT
Reason: Disabled by an administrator.
   See: http://sun.com/msg/SMF-8000-05
   See: man -M /usr/share/man -s 1M lpsched
Impact: 2 services are not running:
        svc:/application/print/rfc1179:default
        svc:/application/print/ipp-listener:default
```

The `-x` option provides additional information about the service instances that are impacted.

3 Enable the service.

```
# svcadm enable application/print/server
```

▼ How to Repair a Corrupt Repository

This procedure shows how to replace a corrupt repository with a default copy of the repository. When the repository daemon, `svc.configd`, is started, it does an integrity check of the configuration repository. This repository is stored in `/etc/svc/repository.db`. The repository can become corrupted due to one of the following reasons:

- Disk failure
- Hardware bug
- Software bug
- Accidental overwrite of the file

If the integrity check fails, the `svc.configd` daemon writes a message to the console similar to the following:

```
svc.configd: smf(5) database integrity check of:

    /etc/svc/repository.db
```

```
failed.  The database might be damaged or a media error might have
prevented it from being verified.  Additional information useful to
your service provider is in:

  /system/volatile/db_errors

The system will not be able to boot until you have restored a working
database.  svc.startd(1M) will provide a sulogin(1M) prompt for recovery
purposes.  The command:

  /lib/svc/bin/restore_repository

can be run to restore a backup version of your repository. See
http://sun.com/msg/SMF-8000-MY for more information.
```

The `svc.startd` daemon then exits and starts `sulogin` to enable you to perform maintenance.

1 Enter the `root` password at the `sulogin` prompt.

The `sulogin` command enables the `root` user to enter system maintenance mode to repair the system.

2 Run the following command:

```
# /lib/svc/bin/restore_repository
```

Running this command takes you through the necessary steps to restore a non-corrupt backup. SMF automatically takes backups of the repository at key system moments. For more information see "SMF Repository Backups" on page 108.

When started, the `/lib/svc/bin/restore_repository` command displays a message similar to the following:

```
See http://sun.com/msg/SMF-8000-MY for more information on the use of
this script to restore backup copies of the smf(5) repository.

If there are any problems which need human intervention, this script will
give instructions and then exit back to your shell.
```

If the system that you are recovering is not a local zone, the script explains how to remount the `/` and `/usr` file systems with read and write permissions to recover the databases. The script exits after printing these instructions. Follow the instructions, paying special attention to any errors that might occur.

After the `root` (`/`) file system is mounted with write permissions, or if the system is a local zone, you are prompted to select the repository backup to restore:

```
The following backups of /etc/svc/repository.db exists, from
oldest to newest:

... list of backups ...
```

Backups are given names, based on type and the time the backup was taken. Backups beginning with `boot` are completed before the first change is made to the repository after system boot.

Backups beginning with `manifest_import` are completed after `svc:/system/manifest-import:default` finishes its process. The time of the backup is given in *YYYYMMDD_HHMMSS* format.

3 Enter the appropriate response.

Typically, the most recent backup option is selected.

```
Please enter either a specific backup repository from the above list to
restore it, or one of the following choices:

        CHOICE            ACTION
        ----------------  --------------------------------------------
        boot              restore the most recent post-boot backup
        manifest_import   restore the most recent manifest_import backup
        -seed-            restore the initial starting repository  (All
                            customizations will be lost, including those
                            made by the install/upgrade process.)
        -quit-            cancel script and quit

Enter response [boot]:
```

If you press Enter without specifying a backup to restore, the default response, enclosed in `[]` is selected. Selecting `-quit-` exits the `restore_repository` script, returning you to your shell prompt.

Note – Selecting `-seed-` restores the `seed` repository. This repository is designed for use during initial installation and upgrades. Using the `seed` repository for recovery purposes should be a last resort.

After the backup to restore has been selected, it is validated and its integrity is checked. If there are any problems, the `restore_repository` command prints error messages and prompts you for another selection. Once a valid backup is selected, the following information is printed, and you are prompted for final confirmation.

```
After confirmation, the following steps will be taken:

svc.startd(1M) and svc.configd(1M) will be quiesced, if running.
/etc/svc/repository.db
    -- renamed --> /etc/svc/repository.db_old_YYYYMMDD_HHMMSS
/system/volatile/db_errors
    -- copied --> /etc/svc/repository.db_old_YYYYMMDD_HHMMSS_errors
repository_to_restore
    -- copied --> /etc/svc/repository.db
and the system will be rebooted with reboot(1M).

Proceed [yes/no]?
```

4 Type yes to remedy the fault.

The system reboots after the `restore_repository` command executes all of the listed actions.

How to Boot Without Starting Any Services

If problems with starting services occur, sometimes a system will hang during the boot. This procedure shows how to troubleshoot this problem.

1. **Boot without starting any services.**

   This command instructs the `svc.startd` daemon to temporarily disable all services and start `sulogin` on the console.

   ```
   ok boot -m milestone=none
   ```

2. **Log in to the system as `root`.**

3. **Enable all services.**

   ```
   # svcadm milestone all
   ```

4. **Determine where the boot process is hanging.**

   When the boot process hangs, determine which services are not running by running `svcs -a`. Look for error messages in the log files in `/var/svc/log`.

5. **After fixing the problems, verify that all services have started.**

   a. **Verify that all needed services are online.**

      ```
      # svcs -x
      ```

   b. **Verify that the `console-login` service dependencies are satisfied.**

      This command verifies that the `login` process on the console will run.

      ```
      # svcs -l system/console-login:default
      ```

6. **Continue the normal booting process.**

How to Force an `sulogin` Prompt If the `system/filesystem/local:default` Service Fails During Boot

Local file systems that are not required to boot the system are mounted by the `svc:/system/filesystem/local:default` service. When any of those file systems are unable to be mounted, the service enters a maintenance state. System startup continues, and any services which do not depend on `filesystem/local` are started. Services which require `filesystem/local` to be online before starting through dependencies are not started.

To change the configuration of the system so that a `sulogin` prompt appears immediately after the service fails instead of allowing system startup to continue, follow the procedure below.

1 Modify the `system/console-login` service.

```
# svccfg -s svc:/system/console-login
svc:/system/console-login> addpg site,filesystem-local dependency
svc:/system/console-login> setprop site,filesystem-local/entities = fmri: svc:/system/filesystem/local

svc:/system/console-login> setprop site,filesystem-local/grouping = astring: require_all

svc:/system/console-login> setprop site,filesystem-local/restart_on = astring: none

svc:/system/console-login> setprop site,filesystem-local/type = astring: service

svc:/system/console-login> end
```

2 Refresh the service.

```
# svcadm refresh console-login
```

Troubleshooting When a failure occurs with the `system/filesystem/local:default` service, the `svcs -vx` command should be used to identify the failure. After the failure has been fixed, the following command clears the error state and allows the system boot to continue: `svcadm clear filesystem/local`.

◆ ◆ ◆ CHAPTER 8

Using the Fault Manager

The Oracle Solaris OS includes an architecture for building and deploying systems and services that are capable of predictive self healing. The service that is the core of the Fault Management Architecture (FMA) receives data related to hardware and software errors, automatically diagnoses the underlying problem, and responds by trying to take faulty components offline.

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

- "Fault Management Overview" on page 143
- "Notification of Faults and Defects" on page 145
- "Displaying Information About Faults or Defects" on page 145
- "Repairing Faults or Defects" on page 149
- "Fault Management Log Files" on page 151
- "Fault Statistics" on page 151

Fault Management Overview

The Oracle Solaris Fault Management feature provides an architecture for building resilient error handlers, structured error telemetry, automated diagnostic software, response agents, and structured messaging. Many parts of the software stack participate in Fault Management, including the CPU, memory and I/O subsystems, Oracle Solaris ZFS, an increasing set of device drivers, and other management stacks.

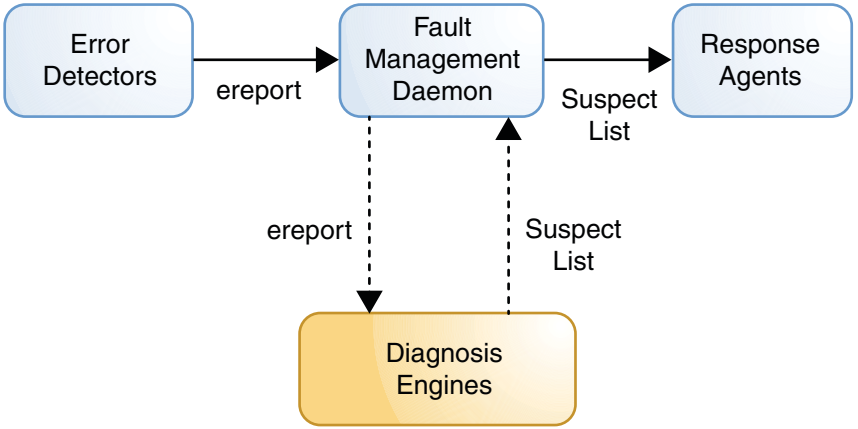
FMA is intended to help with problems that can occur on an Oracle Solaris system. The problem could be a fault, meaning that something that used to work but no longer does. The problem could be a defect, meaning that it never worked correctly. In general, hardware can experience both faults and defects. However, most software problems are defects or are caused by configuration issues.

At a high level, the Fault Management stack contains error detectors, diagnosis engines, and response agents. *Error detectors*, as the name suggests, detect errors in the system and perform any immediate, required handling. Error detectors issue well-defined error reports, or *ereports*,

to a diagnosis engine. A *diagnosis engine* interprets ereports and determines whether a fault or defect is present in the system. When such a determination is made, the diagnosis engine issues a *suspect list* that describes the resource or set of resources that might be the cause of the problem. The resource might or might not have an associated field-replaceable unit (FRU), a label, or an Automatic System Reconfiguration Unit (ASRU). An ASRU may be immediately removed from service to mitigate the problem until the FRU is replaced.

When the suspect list includes multiple suspects, for example, if the diagnosis engine cannot isolate a single suspect, the suspects are assigned a probability as to each suspect being the key suspect. The probabilities in this list add up to 100 percent. Suspect lists are interpreted by response agents. A *response agent* attempts to take some action based on the suspect list. Responses include logging messages, taking CPU strands offline, retiring memory pages, and retiring I/O devices.

Error detectors, diagnosis engines, and response agents are connected by the Fault Manager daemon, `fmd`, which acts as a multiplexor between the various components, as shown in the following figure.



The Fault Manager daemon is itself a service under SMF control. The service is enabled by default and controlled just like any other SMF service. See the `smf(5)` man page for more information.

The FMA and SMF services interact with each other when appropriate. Certain hardware problems can cause services to be stopped or restarted by SMF. Also, certain SMF errors cause FMA to report a defect.

Notification of Faults and Defects

Often, the first interaction with the Fault Manager daemon is a system message indicating that a fault or defect has been diagnosed. Messages are sent to both the console and the `/var/adm/messages` file. All messages from the Fault Manager daemon use the following format:

```
1    SUNW-MSG-ID: SUN4V-8001-8H, TYPE: Fault, VER: 1, SEVERITY: Minor
2    EVENT-TIME: Wed Aug 24 21:56:03 UTC 2011
3    PLATFORM: SUNW,T5440, CSN: -, HOSTNAME: bur419-61
4    SOURCE: cpumem-diagnosis, REV: 1.7
5    EVENT-ID: 7b83c87c-78f6-6a8e-fa2b-d0cf16834049
6    DESC: The number of integer register errors associated with this thread has
7    exceeded acceptable levels.
8    AUTO-RESPONSE: The fault manager will attempt to remove the affected thread
9    from service.
10   IMPACT: System performance may be affected.
11   REC-ACTION: Use 'fmadm faulty' to provide a more detailed view of this
12   event. Please refer to the associated reference document at
13   http://sun.com/msg/SUN4V-8001-8H for the latest service procedures and
14   policies regarding this diagnosis.
```

When notified of a diagnosed problem, always consult the recommended knowledge article for additional details. See line 13 above for an example. The knowledge article might contain additional actions that you or a service provider should take beyond those listed on line 11.

Notification of Fault Manager error events can be configured by using the Simple Network Management Protocol (SNMP) or the Simple Mail Transfer Protocol (SMTP). See "How to Set Up Email Notification of SMF Transition Events" on page 122 for instructions.

In addition, Oracle Auto Service Request can be configured to automatically request Oracle service when specific hardware problems occur. See the Oracle Auto Service Request product page for information about this feature. The documentation link on this page provides links to *Oracle ASR Quick Installation Guide* and *Oracle ASR Installation and Operations Guide*.

Displaying Information About Faults or Defects

The preferred method to display fault or defect information and determine the FRUs involved is the `fmadm faulty` command. However, the `fmdump` command is also supported. `fmdump` is often used to display a historical log of problems on the system, and `fmadm faulty` is used to display the active problems.

Caution – Do not base administrative action on the output of the `fmdump` command, but rather on the `fmadm faulty` output. The log files can contain error statements, which should not be considered faults or defects.

How to Display Information About Faulty Components

1. **Become an administrator.**

   For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2. **Display information about the components.**

   ```
   # fmadm faulty
   ```

   See the following examples for a description of the text generated.

Example 8–1 `fmadm` Output With One Faulty CPU

```
1    # fmadm faulty
2    --------------- ------------------------------------  -------------- ---------
3    TIME            EVENT-ID                              MSG-ID         SEVERITY
4    --------------- ------------------------------------  -------------- ---------
5    Aug 24 17:56:03 7b83c87c-78f6-6a8e-fa2b-d0cf16834049  SUN4V-8001-8H  Minor
6
7    Host        : bur419-61
8    Platform    : SUNW,T5440        Chassis_id  : BEL07524BN
9    Product_sn  : BEL07524BN
10
11   Fault class : fault.cpu.ultraSPARC-T2plus.ireg
12   Affects     : cpu:///cpuid=0/serial=1F95806CD1421929
13                     faulted and taken out of service
14   FRU         : "MB/CPU0" (hc://:product-id=SUNW,T5440:server-id=bur419-61:\
15                 serial=3529:part=541255304/motherboard=0/cpuboard=0)
16                     faulty
17   Serial ID.  : 3529
18                 1F95806CD1421929
19
20   Description : The number of integer register errors associated with this thread
21                 has exceeded acceptable levels.
22
23   Response    : The fault manager will attempt to remove the affected thread from
24                 service.
25
26   Impact      : System performance may be affected.
27
28   Action      : Use 'fmadm faulty' to provide a more detailed view of this event.
29                 Please refer to the associated reference document at
30                 http://sun.com/msg/SUN4V-8001-8H for the latest service
31                 procedures and policies regarding this diagnosis.
```

Of primary interest is line 14, which shows the data for the impacted FRUs. The more human-readable location string is presented in quotation marks, "MB/CPU0". The quoted value is intended to match the label on the physical hardware. The FRU is also represented in a Fault Management Resource Identifier (FMRI) format, which includes descriptive properties about the system containing the fault, such as its host name and chassis serial number. On platforms that support it, the part number and serial number of the FRU are also included in the FRU's FMRI.

The Affects lines (lines 12 and 13) indicate the components that are affected by the fault and their relative state. In this example, a single CPU strand is affected. It is `faulted and taken out of service`.

Following the FRU description in the `fmadm faulty` command output, line 16 shows the state as `faulty`. The Action section might also include other specific actions instead of, or in addition to, the usual reference to the `fmadm` command.

Example 8–2 `fmadm` Output With Multiple Faults

```
1    # fmadm faulty
2    --------------- ------------------------------------  -------------- -------
3    TIME            EVENT-ID                              MSG-ID         SEVERITY
4    --------------- ------------------------------------  -------------- -------
5    Sep 21 10:01:36 d482f935-5c8f-e9ab-9f25-d0aaafec1e6c  PCIEX-8000-5Y  Major
6
7    Fault class  : fault.io.pci.device-invreq
8    Affects      : dev:///pci@0,0/pci1022,7458@11/pci1000,3060@0
9                   dev:///pci@0,0/pci1022,7458@11/pci1000,3060@1
10                   ok and in service
11                  dev:///pci@0,0/pci1022,7458@11/pci1000,3060@2
12                  dev:///pci@0,0/pci1022,7458@11/pci1000,3060@3
13                    faulty and taken out of service
14   FRU          : "SLOT 2" (hc://.../pciexrc=3/pciexbus=4/pciexdev=0)
15                    repair attempted
16                  "SLOT 3" (hc://.../pciexrc=3/pciexbus=4/pciexdev=1)
17                    acquitted
18                  "SLOT 4" (hc://.../pciexrc=3/pciexbus=4/pciexdev=2)
19                    not present
20                  "SLOT 5" (hc://.../pciexrc=3/pciexbus=4/pciexdev=3)
21                    faulty
22
23    Description  : The transmitting device sent an invalid request.
24
25    Response     : One or more device instances may be disabled
26
27    Impact       : Possible loss of services provided by the device instances
28                   associated with this fault
29
30    Action       : Use 'fmadm faulty' to provide a more detailed view of this event.
31                   Please refer to the associated reference document at
32                   http://sun.com/msg/PCIEX-8000-5Y for the latest service
33                   procedures and policies regarding this diagnosis.
```

Following the FRU description in the `fmadm faulty` command output, line 21 shows the state as `faulty`. Other state values that you might see in other situations include `acquitted` and `repair attempted`, as shown for `SLOT 2` and `SLOT 3` in lines 15 and 17.

Example 8–3 Showing Faults with the `fmdump` Command

Some console messages and knowledge articles might instruct you to use the older `fmdump -v -u UUID` command to display fault information. Although the `fmadm faulty` command is preferred, the `fmdump` command still operates, as shown in the following example:

```
1    % fmdump -v -u 7b83c87c-78f6-6a8e-fa2b-d0cf16834049
2    TIME                 UUID                                 SUNW-MSG-ID EVENT
3    Aug 24 17:56:03.4596 7b83c87c-78f6-6a8e-fa2b-d0cf16834049 SUN4V-8001-8H Diagnosed
4      100%  fault.cpu.ultraSPARC-T2plus.ireg
5
6            Problem in: -
7               Affects: cpu:///cpuid=0/serial=1F95806CD1421929
8                   FRU: hc://:product-id=SUNW,T5440:server-id=bur419-61:\
9                   serial=9999:part=541255304/motherboard=0/cpuboard=0
10              Location: MB/CPU0
```

The information about the affected FRUs is still present, although separated across three lines (lines 8 through 10). The Location string presents the human-readable FRU string. The FRU lines presents the formal FMRI. Note that the severity, descriptive text, and action are not shown with the `fmdump` command, unless you use the `-m` option. See the `fmdump(1M)` man page for more information.

How to Identify Which CPUs Are Offline

- **Display information about the CPUs.**

```
% /usr/sbin/psrinfo
0       faulted   since 05/13/2011 12:55:26
1       on-line   since 05/12/2011 11:47:26
```

The `faulted` state indicates that the CPU has been taken offline by a Fault Management response agent.

How to Display Information About Defective Services

1. **Become an administrator.**

   For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2. **Display information about the defect.**

```
# fmadm faulty
--------------- ------------------------------------  -------------- ---------
TIME            EVENT-ID                              MSG-ID         SEVERITY
--------------- ------------------------------------  -------------- ---------
May 12 22:52:47 915cb64b-e16b-4f49-efe6-de81ff96fce7  SMF-8000-YX    major

Host        : parity
Platform    : Sun-Fire-V40z     Chassis_id  : XG051535088
Product_sn  : XG051535088

Fault class : defect.sunos.smf.svc.maintenance
Affects     : svc:///system/intrd:default
                  faulted and taken out of service
Problem in  : svc:///system/intrd:default
```

```
                    faulted and taken out of service

Description : A service failed - it is restarting too quickly.

Response    : The service has been placed into the maintenance state.

Impact      : svc:/system/intrd:default is unavailable.

Action      : Run 'svcs -xv svc:/system/intrd:default' to determine the
              generic reason why the service failed, the location of any
              logfiles, and a list of other services impacted. Please refer to
              the associated reference document at
              http://sun.com/msg/SMF-8000-YX for the latest service procedures
              and policies regarding this diagnosis.
```

3 Display information about the defective service.

Follow the instructions given in the Action section in the `fmadm` output.

```
# svcs -xv svc:/system/intrd:default
svc:/system/intrd:default (interrupt balancer)
 State: maintenance since Wed May 12 22:52:47 2010
Reason: Restarting too quickly.
   See: http://sun.com/msg/SMF-8000-YX
   See: man -M /usr/share/man -s 1M intrd
   See: /var/svc/log/system-intrd:default.log
Impact: This service is not running.
```

Refer to the knowledge article, SMF-8000-YX, for further instructions on fixing this problem.

Repairing Faults or Defects

After Fault Management has faulted a component in your system, you will want to repair it. A repair can happen in one of two ways: implicitly or explicitly.

An *implicit repair* can occur when the faulty component is replaced or removed, provided the component has serial number information that the Fault Manager daemon can track. On many SPARC based systems, serial number information is included in the FMRIs so that the Fault Manager daemon can determine when components have been removed from operation, either through replacement or other means (for example, *blacklisting*). When such detections occur, the Fault Manager daemon no longer displays the affected resource in `fmadm faulty` output. The resource is maintained in the daemon's internal resource cache until the fault event is 30 days old, at which point it is purged.

Implicit repairs do not apply to all systems. Sometimes, even though there is a chassis-id in the FMRIs, no FRU serial number information is available. So the Fault Manager daemon cannot detect a FRU replacement, requiring an *explicit repair*.

The `fmadm` command is used to explicitly mark a fault as repaired. Four syntaxes are associated with repairs for this command:

- `fmadm replaced` *fmri* | *label*
- `fmadm repaired` *fmri* | *label*
- `fmadm acquit` *fmri* | *label*
- `fmadm acquit` *uuid* [*fmri* | *label*]

Although these four commands can take FMRIs and UUIDs as arguments, the preferred argument to use is the label. If a FRU has multiple faults against it, you want to replace the FRU only one time. If you issue the `fmadm replaced` command against the Label, the FRU is reflected as such in any outstanding cases.

`fmadm replaced` Command

You can use the `fmadm replaced` command to indicate that the suspect FRU has been replaced or removed.

If the system automatically discovers that a FRU has been replaced (the serial number has changed), then this discovery is treated in the same way as if `fmadm replaced` had been typed on the command line. The `fmadm replaced` command is not allowed if `fmd` can automatically confirm that the FRU has not been replaced (the serial number has not changed).

If the system automatically discovers that a FRU has been removed but not replaced, then the current behavior is unchanged: The suspect is displayed as `not present`, but is not considered to be permanently removed until the fault event is 30 days old, at which point it is purged.

`fmadm repaired` Command

You can use the `fmadm repaired` command when some physical repair has been carried out to resolve the problem, other than replacing a FRU. Examples of such repairs include reseating a card or straightening a bent pin.

`fmadm acquit` Command

Often you use the `acquit` option when you determine that the resource was not the cause. Acquittal can also happen implicitly when additional error events occur, and the diagnosis gets refined.

Replacement takes precedence over repair, and both replacement and repair take precedence over acquittal. Thus, you can acquit a component and then subsequently repair it, but you cannot acquit a component that has already been repaired.

A case is considered repaired (moves into the `FMD_CASE_REPAIRED` state and a `list.repaired` event is generated) when either its UUID is acquitted, or all suspects have been either repaired, replaced, removed, or acquitted.

Usually `fmd` automatically acquits a suspect in a multi-element suspect list, or Support Services gives you instructions to perform a manual acquittal. You would only want to acquit by FMRI or label if you determined that the resource was not guilty in all current cases in which it is a suspect. However, to allow a FRU to be manually acquitted in one case while remaining a suspect in all others, the following option enables you to specify both UUID and FMRI, or UUID and label:

`fmadm acquit` *uuid* [*fmri*|*label*]

Fault Management Log Files

The Fault Manager daemon, `fmd`, records information in several log files. The log files are stored in `/var/fm/fmd` and are viewed by using the `fmdump` command. See the `fmdump(1M)` man page for more information.

- The `errlog` log file records inbound telemetry information which consists of ereports.
- Informational events are recorded in two log files. `infolog_hival` is for high-value events, and `infolog` collects all other informational events.
- The `fltlog` log file records fault diagnosis and repair events.

Caution – Do not base administrative action on the contents of the log files, but rather on the `fmadm faulty` output. The log files can contain error statements, which should not be considered faults or defects.

The log files are automatically rotated. See the `logadm(1M)` man page for more information.

Fault Statistics

The Fault Manager daemon, `fmd`, and many of its modules track statistics. The `fmstat` command reports those statistics. Without options, `fmstat` gives a high-level overview of the events, processing times, and memory usage of the loaded modules. For example:

```
# fmstat
module                 ev_recv ev_acpt wait  svc_t  %w  %b  open solve  memsz  bufsz
cpumem-retire                1       0  0.0  403.5   0   0     0     0   419b      0
disk-transport               0       0  0.0  500.6   0   0     0     0    32b      0
eft                          0       0  0.0    4.8   0   0     0     0   1.4M    43b
fmd-self-diagnosis           0       0  0.0    4.7   0   0     0     0      0      0
```

```
io-retire                0       0  0.0    4.5   0   0     0     0      0       0
snmp-trapgen             0       0  0.0    4.5   0   0     0     0    32b       0
sysevent-transport       0       0  0.0 1444.4   0   0     0     0      0       0
syslog-msgs              0       0  0.0    4.5   0   0     0     0      0       0
zfs-diagnosis            0       0  0.0    4.7   0   0     0     0      0       0
zfs-retire               0       0  0.0    4.5   0   0     0     0      0       0
```

The fmstat(1M) man page describes each column in this output. Note that the `open` and `solve` columns apply only to Fault Management cases, which are only created and solved by diagnosis engines. These columns are immaterial for other modules, such as response agents.

You may display statistics on an individual module by using the `-m module` option. This syntax is commonly used with the `-z` option to suppress zero-valued statistics. For example:

```
# fmstat -z -m cpumem-retire
  NAME VALUE            DESCRIPTION
  cpu_flts 1            cpu faults resolved
```

This example shows that the `cpumem-retire` response agent has successfully processed a request to take a CPU offline.

◆ ◆ ◆ CHAPTER 9

Managing System Information (Tasks)

This chapter describes the tasks that are required to display and change the most common system information.

This is a list of the information that is in this chapter:

- "What's New in Displaying and Changing System Information" on page 153
- "Displaying System Information (Task Map)" on page 154
- "Changing System Information (Task Map)" on page 160

This chapter does not cover information about resource management that enables you to allocate, monitor, and control system resources in a flexible way. For information about managing system resources with resource management, see Chapter 1, "Introduction to Resource Management," in *Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management*.

What's New in Displaying and Changing System Information

Support for Administratively Provided `driver.conf` Files

System-provided `driver.conf` files can be supplemented with administratively provided `driver.conf` files. Vendor-provided (system-provided) driver data is preserved in the root file system, while administratively provided driver configuration is stored separately in a new `/etc/driver/drv` directory. The format of an administratively provided `driver.conf` file is identical to a system-provided `driver.conf` file.

At boot time, and whenever a `driver.conf` file for a driver is searched for and loaded, the system will also check for the driver in the `driver.conf` file within the `/etc/driver/drv` directory. If found, the system automatically merges the vendor-provider `driver.conf` files

with the local, administratively provided `driver.conf` files. The driver's view of the system properties consists of these merged properties. Therefore, no driver changes are necessary.

Note that vendor-provided `driver.conf` files that are in the `/kernel` and `/platform` directories can no longer be edited. If you need to supplement a driver's configuration, instead of editing the vendor-provided `driver.conf` file, add a corresponding `driver.conf` file to the local `/etc/driver/drv` directory, then customize that file.

To display the merged properties, use the `prtconf` command. Note that the `prtconf` command has a new `-u` option. This option enables you to display both the original and updated property values for a driver. For instructions, see "How to Display Default and Customized Property Values for a Device" on page 157.

For more information, see the `driver(4)` and `driver.conf(4)` man pages.

Displaying System Information (Task Map)

Task	Description	For Instructions
Display a system's release information.	Display the contents of the `/etc/release` file to identify the Oracle Solaris release version.	"How to Display a System's Release Information" on page 155
Display a system's host ID number.	Use the `hostid` command to display your system's host id.	"How to Display a System's Host ID Number" on page 156
Display a system's product name.	You can use the `prtconf -b` command to display the product name of a system.	"How to Display a System's Product Name" on page 156
Display a system's installed memory.	Use the `prtconf` command to display information about your system's installed memory.	"How to Display a System's Installed Memory" on page 156
Display the original and default values for a device.	Use the `prtconf` command with the `-u` option to display both the default and updated property values for a device.	"How to Display Default and Customized Property Values for a Device" on page 157
Display a system's date and time.	Use the `date` command to display your system's date and time.	"How to Display the Date and Time" on page 158

Task	Description	For Instructions
Display a system's physical processor type.	Use the `psrinfo -p` command to list the total number of physical processors on a system. Use the `psrinfo -pv` command to list all physical processors on a system and the virtual processors that are associated with each physical processor.	"How to Display a System's Physical Processor Type" on page 158
Display a system's logical processor type.	Use the `psrinfo -v` command to display a system's logical processor type.	"How to Display a System's Logical Processor Type" on page 159

Displaying System Information

The following table describes commands that enable you to display general system information.

TABLE 9–1 Commands for Displaying System Information

Command	System Information Displayed	Man Page
`date`	Date and time	`date(1)`
`hostid`	Host ID number	`hostid(1)`
`isainfo`	The number of bits supported by *native* applications on the running system, which can be passed as a token to scripts	`isainfo(1)`
`isalist`	Processor type for x86 based systems	`psrinfo(1M)`
`prtconf`	System configuration information, installed memory, device properties, and product name	`prtconf(1M)`
`psrinfo`	Processor type	`psrinfo(1M)`
`uname`	Operating system name, release, version, node name, hardware name, and processor type	`uname(1)`

▼ How to Display a System's Release Information

- **Display the contents of the `/etc/release` file to identify your release version.**

```
$ cat /etc/release
                          Oracle Solaris Nevada Next Development snv_146 x86
```

```
                Copyright (c) 2010, Oracle and/or its affiliates.  All Rights Reserved.
                          Assembled 29 July 2010
```

How to Display a System's Host ID Number

- **To display the host ID number in hexadecimal format, use the `hostid` command.**

Example 9–1 Displaying a System's Host ID Number

The following example shows sample output from the `hostid` command.

```
$ hostid
80a5d34c
```

How to Display a System's Product Name

The `-b` option to the `prtconf` command enables you to display a system's product name. For more information about this feature, see the `prtconf(1M)` man page.

- **To display the product name for your system, use the `prtconf` command with the `-b` option, as follows:**

```
$ prtconf -b
```

Example 9–2 Displaying a System's Product Name

This example shows sample output from the `prtconf -b` command.

```
$ prtconf -b
name:  SUNW,Sun-Fire-T200
banner-name:  Sun Fire T200
compatible: 'sun4v'
```

This example shows sample output from the `prtconf -vb` command.

```
$ prtconf -vb
name:  SUNW,Sun-Fire-T200
banner-name:  Sun Fire T200
compatible: 'sun4v'
idprom:  01840014.4f1de8da.00000000.1de8dade.00000000.00000000.00000000.00000000
openprom model:  SUNW,4.30.4.a
openprom version: 'OBP 4.30.4.a 2010/01/06 14:56'
```

How to Display a System's Installed Memory

- **To display the amount of memory that is installed on your system, use the `prtconf` command.**

Example 9–3 Displaying a System's Installed Memory

The following example shows sample output from the `prtconf` command. The `grep Memory` command selects output from the `prtconf` command to display memory information only.

```
$ prtconf | grep Memory
Memory size: 65408 Megabytes
```

How to Display Default and Customized Property Values for a Device

To display both the default and customized property values for devices, use the `prtconf` command with the `-u` option. For more information about this option, see the `prtconf(1M)` man page.

- **Display the default and customized properties of a `driver.conf` file.**

  ```
  $ prtconf -u
  ```

  The output of the `prtconf -u` command displays the default and customized properties for all of the drivers that are on the system.

Example 9–4 Displaying Default

This example shows the default and custom properties for the `bge.conf` file. Note that vendor-provided configuration files are located in the `/kernel` and `/platform` directories, while the corresponding modified driver configuration files are located in the `/etc/driver/drv` directory.

```
$ prtconf -u

.
.
.
pci108e,534d (pci14e4,16a7), instance #0
          System software properties:
            name='bge-known-subsystems' type=int items=16
            name='bge-rx-rings' type=int items=1
                  value=00000010
            name='bge-tx-rings' type=int items=1
                  value=00000002 <---- system merged value 2
          Admin global properties:
             name='bge-tx-rings' type=int items=1
                  value=00000002 <---- admin value is 2
          Vendor global properties:
             name='bge-tx-rings' type=int items=1
                       value=00000001 <---- vendor value is 1
.
.
.
```

See Also For more information, see the `driver(4)` and `driver.conf(4)` man pages.

For instructions on how to create administratively provided configuration files, see Chapter 5, "Managing Devices (Overview/Tasks)," in *Oracle Solaris Administration: Devices and File Systems*.

▼ How to Display the Date and Time

- **To display the current date and time according to your system clock, use the `date` command.**

Example 9–5 Displaying the Date and Time

The following example shows sample output from the `date` command.

```
$ date
Mon Sep 13 17:32:59 MST 2010
$
```

Identifying Information About Chip Multithreading Features

The `psrinfo` command has been modified to provide information about physical processors, in addition to information about virtual processors. This enhanced functionality has been added to identify chip multithreading (CMT) features. The new `-p` option reports the total number of physical processors that are in a system. Using the `psrinfo -pv` command will list all the physical processors that are in the system, as well as the virtual processors that are associated with each physical processor. The default output of the `psrinfo` command continues to display the virtual processor information for a system.

For more information, see the `psrinfo(1M)` man page.

For information about the procedures that are associated with this feature, see "How to Display a System's Physical Processor Type" on page 158.

▼ How to Display a System's Physical Processor Type

- **Use the `psrinfo -p` command to display the total number of physical processors on a system.**

```
$ psrinfo -p
1
```

Use the `psrinfo -pv` command to display information about each physical processor on a system, and the virtual processor that is associated with each physical processor.

```
$ psrinfo -pv
The UltraSPARC-IV physical processor has 2 virtual processors (8, 520)
The UltraSPARC-IV physical processor has 2 virtual processors (9, 521)
The UltraSPARC-IV physical processor has 2 virtual processors (10, 522)
The UltraSPARC-IV physical processor has 2 virtual processors (11, 523)
The UltraSPARC-III+ physical processor has 1 virtual processor (16)
The UltraSPARC-III+ physical processor has 1 virtual processor (17)
The UltraSPARC-III+ physical processor has 1 virtual processor (18)
The UltraSPARC-III+ physical processor has 1 virtual processor (19)
```

When you use the `psrinfo -pv` command on an x86 based system, the following output is displayed:

```
$ psrinfo -pv
The i386 physical processor has 2 virtual processors (0, 2)
The i386 physical processor has 2 virtual processors (1, 3)
```

How to Display a System's Logical Processor Type

- **Use the `psrinfo -v` command to display information about a system's processor type.**

  ```
  $ psrinfo -v
  ```

  On an x86 based system, use the `isalist` command to display the virtual processor type.

  ```
  $ isalist
  ```

Example 9–6 SPARC: Displaying a System's Processor Type

This example shows how to display information about a SPARC based system's processor type.

```
$ psrinfo -v
Status of virtual processor 28 as of: 09/13/2010 14:07:47
  on-line since 04/08/2010 21:27:56.
  The sparcv9 processor operates at 1400 MHz,
        and has a sparcv9 floating point processor.
Status of virtual processor 29 as of: 09/13/2010 14:07:47
  on-line since 04/08/2010 21:27:56.
  The sparcv9 processor operates at 1400 MHz,
        and has a sparcv9 floating point processor.
```

Example 9–7 x86: Displaying a System's Processor Type

This example shows how to display information about an x86 based system's processor type.

```
$ isalist
pentium_pro+mmx pentium_pro pentium+mmx pentium i486 i386 i86
```

Changing System Information (Task Map)

Task	Directions	For Instructions
Manually set a system's date and time.	Manually set your system's date and time by using the `date` *mmddHHMM[[cc]yy]* command-line syntax.	"How to Manually Set a System's Date and Time" on page 160
Set up a message-of-the-day.	Set up a message-of-the-day on your system by editing the `/etc/motd` file.	"How to Set Up a Message-Of-The-Day" on page 161
Change a system's identity.	Change your system's identity by setting the `config/nodename` SMF property for the `svc:system/identity:node` service.	"How to Change a System's Identity (nodename)" on page 162

Changing System Information

This section describes commands that enable you to change general system information.

▼ How to Manually Set a System's Date and Time

1 **Become an administrator.**

2 **Enter the new date and time.**

`# date` *mmddHHMM[[cc]yy]*

mm Month, using two digits.

dd Day of the month, using two digits.

HH Hour, using two digits and a 24-hour clock.

MM Minutes, using two digits.

cc Century, using two digits.

yy Year, using two digits.

See the `date(1)` man page for more information.

3 **Verify that you have reset your system's date correctly by using the `date` command with no options.**

Example 9–8 Manually Setting a System's Date and Time

The following example shows how to use the `date` command to manually set a system's date and time.

```
# date
Monday, September 13. 2010 02:00:16 PM MDT
# date 0921173404
Thu Sep 17:34:34 MST 2010
```

▼ How to Set Up a Message-Of-The-Day

Edit the message-of-the-day file, `/etc/motd`, to include announcements or inquiries to all users of a system when they log in. Use this feature sparingly, and edit this file regularly to remove obsolete messages.

1 Become the root role.

```
$ su -
Password:
#
```

Note – This method works whether `root` is a user or a role.

---

2 Edit the `/etc/motd` file and add a message of your choice.

Edit the text to include the message that will be displayed during user login. Include spaces, tabs, and carriage returns.

3 Verify the changes by displaying the contents of the `/etc/motd` file.

```
$ cat /etc/motd
Welcome to the UNIX Universe. Have a nice day.
```

Example 9–9 Setting Up a Message-Of-The-Day

The default message-of-the-day, which is provided when you install Oracle Solaris software, contains version information. The following example shows an edited `/etc/motd` file that provides information about system availability to each user who logs in.

```
$ cat /etc/motd
The system will be down from 7:00 a.m to 2:00 p.m. on
Saturday, July 7, for upgrades and maintenance.
Do not try to access the system during those hours.
Thank you.
```

▼ How to Change a System's Identity (`nodename`)

1. **Become the root role.**

2. **To set the name of a host, specify the `config/nodename` SMF property for the `svc:/system/identity:node` service, as follows:**

   ```
   # svccfg -s svc:/system/identity:node setprop config/nodename = some-name
   ```

◆ ◆ ◆ CHAPTER 10

Managing System Processes (Tasks)

This chapter describes the procedures for managing system processes.

This is a list of the information that is in this chapter:

- "Managing System Processes (Task Map)" on page 163
- "Managing Process Class Information (Task Map)" on page 173

Managing System Processes (Task Map)

Task	Description	For Instructions
List processes.	Use the `ps` command to list all the processes on a system.	"How to List Processes" on page 167
Display information about processes.	Use the `pgrep` command to obtain the process IDs for processes that you want to display more information about.	"How to Display Information About Processes" on page 168
Control processes.	Locate processes by using the `pgrep` command. Then, use the appropriate `pcommand` (`/proc`) to control the process. See Table 10–3 for a description of the (`/proc`) commands.	"How to Control Processes" on page 169
Kill a process.	Locate a process, either by process name or process ID. You can use either the `pkill` or `kill` commands to terminate the process.	"How to Terminate a Process (`pkill`)" on page 170 "How to Terminate a Process (`kill`)" on page 171

Commands for Managing System Processes

The following table describes the commands for managing system processes.

TABLE 10–1 Commands for Managing Processes

Command	Description	Man Page
`ps`, `pgrep`, `prstat`, `pkill`	Checks the status of active processes on a system, as well as displays detailed information about the processes.	`ps`(1), `pgrep`(1), and`prstat`(1M)
`pkill`	Functions identically to `pgrep` but finds or signals processes by name or other attribute and terminates the process. Each matching process is signaled as if by the `kill` command, instead of having its process ID printed.	`pgrep`(1), and `pkill`(1) `kill`(1)
`pargs, preap`	Assists with processes debugging.	`pargs`(1), and `preap`(1)
`dispadmin`	Lists default process scheduling policies.	`dispadmin`(1M)
`priocntl`	Assigns processes to a priority class and manages process priorities.	`priocntl`(1)
`nice`	Changes the priority of a timesharing process.	`nice`(1)
`psrset`	Binds specific process groups to a group of processors rather than to just a single processor.	`psrset`(1M)

Using the `ps` Command

The `ps` command enables you to check the status of active processes on a system, as well as display technical information about the processes. This data is useful for administrative tasks, such as determining how to set process priorities.

Depending on which options you use, the `ps` command reports the following information:

- Current status of the process
- Process ID
- Parent process ID
- User ID
- Scheduling class
- Priority

- Address of the process
- Memory used
- CPU time used

The following table describes some fields that are reported by the `ps` command. Which fields are displayed depend on which option you choose. For a description of all available options, see the `ps(1)` man page.

TABLE 10–2 Summary of Fields in `ps` Reports

Field	Description
`UID`	The effective user ID of the process's owner.
`PID`	The process ID.
`PPID`	The parent process ID.
`C`	The processor xutilization for scheduling. This field is not displayed when the `-c` option is used.
`CLS`	The scheduling class to which the process belongs such as real-time, system, or timesharing. This field is included only with the `-c` option.
`PRI`	The kernel thread's scheduling priority. Higher numbers indicate a higher priority.
`NI`	The process's `nice` number, which contributes to its scheduling priority. Making a process “nicer” means lowering its priority.
`ADDR`	The address of the `proc` structure.
`SZ`	The virtual address size of the process.
`WCHAN`	The address of an event or lock for which the process is sleeping.
`STIME`	The starting time of the process in hours, minutes, and seconds.
`TTY`	The terminal from which the process, or its parent, was started. A question mark indicates that there is no controlling terminal.
`TIME`	The total amount of CPU time used by the process since it began.
`CMD`	The command that generated the process.

Using the `/proc` File System and Commands

You can display detailed information about the processes that are listed in the `/proc` directory by using process commands. The following table lists the `/proc` process commands. The `/proc` directory is also known as the process file system (PROCFS). Images of active processes are stored here by their process ID number.

TABLE 10–3 Process Commands (`/proc`)

Process Command	Description
`pcred`	Displays process credential information
`pfiles`	Reports `fstat` and `fcntl` information for open files in a process
`pflags`	Prints `/proc` tracing flags, pending signals and held signals, and other status information
`pldd`	Lists the dynamic libraries that are linked into a process
`pmap`	Prints the address space map of each process
`psig`	Lists the signal actions and handlers of each process
`prun`	Starts each process
`pstack`	Prints a hex+symbolic stack trace for each lwp in each process
`pstop`	Stops each process
`ptime`	Times a process by using microstate accounting
`ptree`	Displays the process trees that contain the process
`pwait`	Displays status information after a process terminates
`pwdx`	Displays the current working directory for a process

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

The process tools are similar to some options of the `ps` command, except that the output that is provided by these commands is more detailed.

In general, the process commands do the following:

- Display more information about processes, such as `fstat` and `fcntl`, working directories, and trees of parent and child processes.
- Provide control over processes by allowing users to stop or resume them.

Managing Processes With Process Commands (`/proc`)

You can display detailed, technical information about processes or control active processes by using some of the process commands. Table 10–3 lists some of the `/proc` commands.

If a process becomes trapped in an endless loop, or if the process takes too long to execute, you might want to stop (kill) the process. For more information about stopping processes using the `kill` or the `pkill` command, see Chapter 10, "Managing System Processes (Tasks)."

The /proc file system is a directory hierarchy that contains additional subdirectories for state information and control functions.

The /proc file system also provides an xwatchpoint facility that is used to remap read-and-write permissions on the individual pages of a process's address space. This facility has no restrictions and is MT-safe.

Debugging tools have been modified to use /proc's xwatchpoint facility, which means that the entire xwatchpoint process is faster.

The following restrictions have been removed when you set xwatchpoints by using the dbx debugging tool:

- Setting xwatchpoints on local variables on the stack due to SPARC based system register windows.
- Setting xwatchpoints on multithreaded processes.

For more information, see the proc(4), and mdb(1) man pages.

How to List Processes

- **Use the ps command to list all the processes on a system.**

  ```
  $ ps [-efc]
  ```

  ps Displays only the processes that are associated with your login session.

  -ef Displays full information about all the processes that are being executed on the system.

  -c Displays process scheduler information.

Example 10–1 Listing Processes

The following example shows output from the ps command when no options are used.

```
$ ps
   PID TTY      TIME COMD
  1664 pts/4    0:06 csh
  2081 pts/4    0:00 ps
```

The following example shows output from the ps -ef command. This output shows that the first process that is executed when the system boots is sched (the swapper) followed by the init process, pageout, and so on.

```
$ ps -ef
 UID   PID  PPID   C    STIME TTY         TIME CMD
    root     0     0   0 18:04:04 ?           0:15 sched
    root     5     0   0 18:04:03 ?           0:05 zpool-rpool
```

```
   root     1     0   0 18:04:05 ?           0:00 /sbin/init
   root     2     0   0 18:04:05 ?           0:00 pageout
   root     3     0   0 18:04:05 ?           2:52 fsflush
   root     6     0   0 18:04:05 ?           0:02 vmtasks
daemon   739     1   0 19:03:58 ?           0:00 /usr/lib/nfs/nfs4cbd
   root     9     1   0 18:04:06 ?           0:14 /lib/svc/bin/svc.startd
   root    11     1   0 18:04:06 ?           0:45 /lib/svc/bin/svc.configd
daemon   559     1   0 18:04:49 ?           0:00 /usr/sbin/rpcbind
netcfg    47     1   0 18:04:19 ?           0:01 /lib/inet/netcfgd
 dladm    44     1   0 18:04:17 ?           0:00 /sbin/dlmgmtd
netadm    51     1   0 18:04:22 ?           0:01 /lib/inet/ipmgmtd
   root   372   338   0 18:04:43 ?           0:00 /usr/lib/hal/hald-addon-cpufreq
   root    67     1   0 18:04:30 ?           0:02 /lib/inet/in.mpathd
   root   141     1   0 18:04:38 ?           0:00 /usr/lib/pfexecd
netadm    89     1   0 18:04:31 ?           0:03 /lib/inet/nwamd
   root   602     1   0 18:04:50 ?           0:02 /usr/lib/inet/inetd start
   root   131     1   0 18:04:35 ?           0:01 /sbin/dhcpagent
daemon   119     1   0 18:04:33 ?           0:00 /lib/crypto/kcfd
   root   333     1   0 18:04:41 ?           0:07 /usr/lib/hal/hald --daemon=yes
   root   370   338   0 18:04:43 ?           0:00 /usr/lib/hal/hald-addon-network-discovery
   root   159     1   0 18:04:39 ?           0:00 /usr/lib/sysevent/syseventd
   root   236     1   0 18:04:40 ?           0:00 /usr/lib/ldoms/drd
   root   535     1   0 18:04:46 ?           0:09 /usr/sbin/nscd
   root   305     1   0 18:04:40 ?           0:00 /usr/lib/zones/zonestatd
   root   326     1   0 18:04:41 ?           0:03 /usr/lib/devfsadm/devfsadmd
   root   314     1   0 18:04:40 ?           0:00 /usr/lib/dbus-daemon --system
   .
       .
       .
```

▼ How to Display Information About Processes

1 Obtain the process ID of the process that you want to display more information about.

```
# pgrep process
```

where *process* is the name of the process you want to display more information about.

The process ID is displayed in the first column of the output.

2 Display the process information that you need.

```
# /usr/bin/pcommand pid
```

pcommand — Is the (`/proc`) command that you want to run. Table 10–3 lists and describes these commands.

pid — Identifies the process ID.

Example 10–2 Displaying Information About Processes

The following example shows how to use process commands to display more information about a `cron` process.

```
# pgrep cron      1
4780
# pwdx 4780      2
4780:   /var/spool/cron/atjobs
# ptree 4780      3
4780  /usr/sbin/cron
# pfiles 4780      4
4780:   /usr/sbin/cron
  Current rlimit: 256 file descriptors
   0: S_IFCHR mode:0666 dev:290,0 ino:6815752 uid:0 gid:3 rdev:13,2
      O_RDONLY|O_LARGEFILE
      /devices/pseudo/mm@0:null
   1: S_IFREG mode:0600 dev:32,128 ino:42054 uid:0 gid:0 size:9771
      O_WRONLY|O_APPEND|O_CREAT|O_LARGEFILE
      /var/cron/log
   2: S_IFREG mode:0600 dev:32,128 ino:42054 uid:0 gid:0 size:9771
      O_WRONLY|O_APPEND|O_CREAT|O_LARGEFILE
      /var/cron/log
   3: S_IFIFO mode:0600 dev:32,128 ino:42049 uid:0 gid:0 size:0
      O_RDWR|O_LARGEFILE
      /etc/cron.d/FIFO
   4: S_IFIFO mode:0000 dev:293,0 ino:4630 uid:0 gid:0 size:0
      O_RDWR|O_NONBLOCK
   5: S_IFIFO mode:0000 dev:293,0 ino:4630 uid:0 gid:0 size:0
      O_RDWR
```

1. Obtains the process ID for the `cron` process
2. Displays the current working directory for the `cron` process
3. Displays the process tree that contains the `cron` process
4. Displays `fstat` and `fcntl` information

▼ How to Control Processes

1 Obtain the process ID of the process that you want to control.

```
# pgrep process
```

where *process* is the name of the process you want to control.

The process ID displayed in the first column of the output.

2 Use the appropriate process command to control the process.

```
# /usr/bin/pcommand pid
```

pcommand Is the process (`/proc`) command that you want to run. Table 10–3 lists and describes these commands.

pid Identifies the process ID.

3 Verify the process status.

```
# ps -ef | grep pid
```

Terminating a Process (`pkill, kill`)

Sometimes, you might need to stop (kill) a process. The process might be in an endless loop. Or, you might have started a large job that you want to stop before it is completed. You can kill any process that you own. Superuser can kill any process in the system except for those processes with process IDs of `0`, `1`, `2`, `3`, and `4`. Killing these processes most likely will crash the system.

For more information, see the `pgrep(1)` and `pkill(1)` and `kill(1)` man pages.

▼ How to Terminate a Process (`pkill`)

1 To terminate the process of another user, become root.

2 Obtain the process ID for the process that you want to terminate.

```
$ pgrep process
```

where *process* is the name of the process that you want to terminate.

For example:

```
$ pgrep netscape
587
566
```

The process ID is displayed in the output.

Note – To obtain process information about a Sun Ray, use the following commands:

```
# ps -fu user
```

This command lists all user processes.

```
# ps -fu user | grep process
```

This command locates a specific process for a user.

---

3 Terminate the process.

```
$ pkill [signal] process
```

signal When no signal is included in the `pkill` command-line syntax, the default signal that is used is –15 (`SIGKILL`). Using the –9 signal (`SIGTERM`) with the `pkill` command ensures that the process terminates promptly. However, the –9 signal should not be used to kill certain processes, such as a database process, or an LDAP server process. The result is that data might be lost.

process Is the name of the process to stop.

Tip – When using the `pkill` command to terminate a process, first try using the command by itself, without including a signal option. Wait a few minutes to see if the process terminates before using the `pkill` command with the -9 signal.

---

4 Verify that the process has been terminated.

```
$ pgrep process
```

The process you terminated should no longer be listed in the output of the `pgrep` command.

▼ How to Terminate a Process (`kill`)

1 To terminate the process of another user, become root.

2 Obtain the process ID of the process that you want to terminate.

```
# ps -fu user
```

where *user* is the user that you want to display processes for.

The process ID is displayed in the first column of the output.

3 Terminate the process.

```
# kill [signal-number] pid
```

signal When no signal is included in the `kill` command-line syntax, the default signal that is used is –15 (`SIGKILL`). Using the –9 signal (`SIGTERM`) with the `kill` command ensures that the process terminates promptly. However, the –9 signal should not be used to kill certain processes, such as a database process, or an LDAP server process. The result is that data might be lost.

pid Is the process ID of the process that you want to terminate.

Tip – When using the `kill` command to stop a process, first try using the command by itself, without including a signal option. Wait a few minutes to see if the process terminates before using the `kill` command with the -9 signal.

---

4 Verify that the process has been terminated.

```
$ pgrep pid
```

The process you terminated should no longer be listed in the output of the `pgrep` command.

Debugging a Process (`pargs`, `preap`)

The `pargs` command and the `preap` command improve process debugging. The `pargs` command prints the arguments and environment variables that are associated with a live process or core file. The `preap` command removes defunct (zombie) processes. A zombie process has not yet had its exit status claimed by its parent. These processes are generally harmless but can consume system resources if they are numerous. You can use the `pargs` and `preap` commands to examine any process that you have the privileges to examine. As superuser, you can examine any process.

For information about using the `preap` command, see the `preap(1)` man page. For information about the using the `pargs` command, see the `pargs(1)` man page. See also, the `proc(1)` man page.

EXAMPLE 10–3 Debugging a Process (`pargs`)

The `pargs` command solves a long-standing problem of being unable to display with the `ps` command all the arguments that are passed to a process. The following example shows how to use the `pargs` command in combination with the `pgrep` command to display the arguments that are passed to a process.

```
# pargs 'pgrep ttymon'
579:    /usr/lib/saf/ttymon -g -h -p system-name console login:
-T sun -d /dev/console -l
argv[0]: /usr/lib/saf/ttymon
argv[1]: -g
argv[2]: -h
argv[3]: -p
argv[4]: system-name console login:
argv[5]: -T
argv[6]: sun
argv[7]: -d
argv[8]: /dev/console
argv[9]: -l
argv[10]: console
argv[11]: -m
argv[12]: ldterm,ttcompat
548:    /usr/lib/saf/ttymon
argv[0]: /usr/lib/saf/ttymon
```

The following example shows how to use the `pargs -e` command to display the environment variables that are associated with a process.

EXAMPLE 10–3 Debugging a Process (`pargs`) *(Continued)*

```
$ pargs -e 6763
6763: tcsh
envp[0]: DISPLAY=:0.0
```

Managing Process Class Information (Task Map)

Task	Description	For Instructions
Display basic information about process classes.	Use the `priocntl -l` command. to Display process scheduling classes and priority ranges.	"How to Display Basic Information About Process Classes (`priocntl`)" on page 174
Display the global priority of a process.	Use the `ps -ecl` command to display the global priority of a process.	"How to Display the Global Priority of a Process" on page 175
Designate a process priority.	Start a process with a designated priority by using the `priocntl -e -c command.`	"How to Designate a Process Priority (`priocntl`)" on page 176
Change scheduling parameters of a timesharing process.	Use the `priocntl -s -m` command to change scheduling parameters in a timesharing process.	"How to Change Scheduling Parameters of a Timesharing Process (`priocntl`)" on page 177
Change the class of a process.	Use the `priocntl -s -c` command to change the class of a process.	"How to Change the Class of a Process (`priocntl`)" on page 178
Change the priority of a process.	Use the `/usr/bin/nice` command with the appropriate options to lower or raise the priority of a process.	"How to Change the Priority of a Process (`nice`)" on page 179

Managing Process Class Information

The following list identifies the process scheduling classes that can be configured on your system. Also included is the user priority range for the timesharing class.

The possible process scheduling classes are as follows:

- Fair share (`FSS`)
- Fixed (`FX`)
- System (`SYS`)
- Interactive (`IA`)

- Real-time (RT)
- Timesharing (TS)
  - The user-supplied priority ranges from -60 to +60.
  - The priority of a process is inherited from the parent process. This priority is referred to as the *user-mode priority*.
  - The system looks up the user-mode priority in the timesharing dispatch parameter table. Then, the system adds in any `nice` or `priocntl` (user-supplied) priority and ensures a 0–59 range to create a *global priority*.

Changing the Scheduling Priority of Processes (`priocntl`)

The scheduling priority of a process is the priority assigned by the process scheduler, according to scheduling policies. The `dispadmin` command lists the default scheduling policies. For more information, see the `dispadmin`(1M) man page.

You can use the `priocntl` command to assign processes to a priority class and to manage process priorities. For instructions on using the `priocntl` command to manage processes, see "How to Designate a Process Priority (`priocntl`)" on page 176.

▼ How to Display Basic Information About Process Classes (`priocntl`)

- **Display process scheduling classes and priority ranges with the `priocntl -l` command.**

```
$ priocntl -l
```

Example 10–4 Displaying Basic Information About Process Classes (`priocntl`)

The following example shows output from the `priocntl -l` command.

```
# priocntl -l
CONFIGURED CLASSES
==================

SYS (System Class)

TS (Time Sharing)
        Configured TS User Priority Range: -60 through 60

FX (Fixed priority)
        Configured FX User Priority Range: 0 through 60
```

```
IA (Interactive)
        Configured IA User Priority Range: -60 through 60
```

▼ How to Display the Global Priority of a Process

- **Display the global priority of a process by using the `ps` command.**

  ```
  $ ps -ecl
  ```

  The global priority is listed under the `PRI` column.

Example 10–5 Displaying the Global Priority of a Process

The following example shows `ps -ecl` command output. The values in the `PRI` column show that the `pageout` process has the highest priority, while the `sh` process has the lowest priority.

```
$ ps -ecl
 F S    UID   PID  PPID  CLS PRI     ADDR     SZ    WCHAN TTY         TIME CMD
 1 T      0     0     0  SYS  96        ?      0        ?         0:11 sched
 1 S      0     5     0  SDC  99        ?      0        ? ?       0:01 zpool-rp
 0 S      0     1     0   TS  59        ?    688        ? ?       0:00 init
 1 S      0     2     0  SYS  98        ?      0        ? ?       0:00 pageout
 1 S      0     3     0  SYS  60        ?      0        ? ?       2:31 fsflush
 1 S      0     6     0  SDC  99        ?      0        ? ?       0:00 vmtasks
 0 S     16    56     1   TS  59        ?   1026        ? ?       0:01 ipmgmtd
 0 S      0     9     1   TS  59        ?   3480        ? ?       0:04 svc.star
 0 S      0    11     1   TS  59        ?   3480        ? ?       0:13 svc.conf
 0 S      0   162     1   TS  59        ?    533        ? ?       0:00 pfexecd
 0 S      0  1738  1730   TS  59        ?    817        ? pts/ 1     0:00 bash
 0 S      1   852     1   TS  59        ?    851        ? ?       0:17 rpcbind
 0 S     17    43     1   TS  59        ?   1096        ? ?       0:01 netcfgd
 0 S     15    47     1   TS  59        ?    765        ? ?       0:00 dlmgmtd
 0 S      0    68     1   TS  59        ?    694        ? ?       0:01 in.mpath
 0 S      1  1220     1   FX  60        ?    682        ? ?       0:00 nfs4cbd
 0 S     16    89     1   TS  59        ?   1673        ? ?       0:02 nwamd
 0 S      0   146     1   TS  59        ?    629        ? ?       0:01 dhcpagen
 0 S      1   129     1   TS  59        ?   1843        ? ?       0:00 kcfd
 0 S      1  1215     1   FX  60        ?    738        ? ?       0:00 lockd
 0 S      0   829   828   TS  59        ?    968        ? ?       0:00 hald-run
 0 S      0   361     1   TS  59        ?   1081        ? ?       0:01 devfsadm
 0 S      0   879     1   TS  59        ?   1166        ? ?       0:01 inetd
 0 O 119764  1773   880   TS  59        ?    557          cons ole     0:00 ps
 0 S      0   844   829   TS  59        ?    996        ? ?       0:00 hald-add
 0 S      0   895   866   TS  59        ?    590        ? ?       0:00 ttymon
 0 S      0   840     1   TS  59        ?    495        ? ?       0:00 cron
 0 S      0   874     1   TS  59        ?    425        ? ?       0:00 utmpd
 0 S      0  1724   956   TS  59        ?   2215        ? ?       0:00 sshd
 0 S 119764   880     9   TS  59        ?    565        ? cons ole     0:00 csh
 0 S      0   210     1   TS  59        ?   1622        ? ?       0:00 sysevent
 0 S      0   279     1   TS  59        ?    472        ? ?       0:00 iscsid
 0 S      1  1221     1   TS  59        ?   1349        ? ?       0:00 nfsmapid
 1 S      0   374     0  SDC  99        ?      0        ? ?       0:00 zpool-us
 0 S      0  1207     1   TS  59        ?   1063        ? ?       0:00 rmvolmgr
 0 S      0   828     1   TS  59        ?   1776        ? ?       0:03 hald
```

```
 0 S     0   853   829  TS  59       ?    896       ? ?         0:02 hald-add
 0 S     0   373     1  TS  59       ?    985       ? ?         0:00 picld
 0 S     0   299     1  TS  59       ?    836       ? ?         0:00 dbus-dae
 0 S 12524  1730  1725  TS  59       ?    452       ? pts/ 1       0:00 csh
 0 S     0   370     1  TS  59       ?    574       ? ?         0:00 powerd
 0 S     0   264     1  FX  60       ?    637       ? ?         0:00 zonestat
 0 S     0   866     9  TS  59       ?    555       ? ?         0:00 sac
 0 S     0   851   829  TS  59       ?    998       ? ?         0:00 hald-add
 0 S 12524  1725  1724  TS  59       ?   2732       ? ?         0:00 sshd
 0 S     1  1211     1  TS  59       ?    783       ? ?         0:00 statd
 0 S     0  1046     1  TS  59       ?   1770       ? ?         0:13 intrd
 0 S     0   889     1  TS  59       ?   1063       ? ?         0:00 syslogd
 0 S     0  1209     1  TS  59       ?    792       ? ?         0:00 in.ndpd
 0 S     0  1188  1186  TS  59       ?    951       ? ?         0:15 automoun
 0 S     0  1172   829  TS  59       ?    725       ? ?         0:00 hald-add
 0 S     0  1186     1  TS  59       ?    692       ? ?         0:00 automoun
 0 S   101  1739  1738  TS  59       ?    817       ? pts/ 1       0:00 bash
 0 S     0  1199     1  TS  59       ?   1495       ? ?         0:02 sendmail
 0 S     0   956     1  TS  59       ?   1729       ? ?         0:00 sshd
 0 S    25  1192     1  TS  59       ?   1528       ? ?         0:00 sendmail
 0 S     0   934     1  TS  59       ?   6897       ? ?         0:14 fmd
 0 S     0  1131     1  TS  59       ?   1691       ? ?         0:07 nscd
 0 S     1  1181     1  TS  59       ?    699       ? ?         0:00 ypbind
```

▼ How to Designate a Process Priority (`priocntl`)

1 Become the root role.

2 Start a process with a designated priority.

```
# priocntl -e -c class -m user-limit -p pri command-name
```

`-e`	Executes the command.
`-c` *class*	Specifies the class within which to run the process. The valid classes are `TS` (timesharing), `RT` (real time), IA (interactive), FSS (fair share), and FX (fixed priority).
`-m` *user-limit*	When you use the `-p` option with this option, the maximum amount you can raise or lower your priority is also specified.
`-p` *pri command-name*	Enables you specify the relative priority in the RT class for a real-time thread. For a timesharing process, the `-p` option lets you specify the user-supplied priority, which ranges from -60 to +60.

3 Verify the process status.

```
# ps -ecl | grep command-name
```

Example 10–6 Designating a Process Priority (`priocntl`)

The following example shows how to start the `find` command with the highest possible user-supplied priority.

```
# priocntl -e -c TS -m 60 -p 60 find . -name core -print
# ps -ecl | grep find
```

▼ How to Change Scheduling Parameters of a Timesharing Process (`priocntl`)

1 Become the root role.

2 Change the scheduling parameters of a running timesharing process.

`# priocntl -s -m` *user-limit* `[-p` *user-priority*`] -i` *idtype idlist*

`-s`	Lets you set the upper limit on the user priority range and change the current priority.
`-m` *user-limit*	When you use the `-p` option, specifies the maximum amount you can raise or lower the priority.
`-p` *user-priority*	Allows you to designate a priority.
`-i` *xidtype xidlist*	Uses a combination of *xidtype* and *xidlist* to identify the process or processes. The *xidtype* specifies the type of ID, such as the process ID or the user ID. Use *xidlist* to identify a list of process IDs or user IDs.

3 Verify the process status.

`# ps -ecl | grep` *idlist*

Example 10–7 Changing Scheduling Parameters of a Timesharing Process (`priocntl`)

The following example shows how to execute a command with a 500-millisecond time slice, a priority of 20 in the RT class, and a global priority of 120.

```
# priocntl -e -c RT -m 500 -p 20 myprog
# ps -ecl | grep myprog
```

▼ How to Change the Class of a Process (`priocntl`)

1 (Optional) Become the root role.

2 Change the class of a process.

```
# priocntl -s -c class -i idtype idlist
```

`-s`	Lets you set the upper limit on the user priority range and change the current priority.
`-c` *class*	Specifies the class, `TS` for time-sharing or `RT` `for real-time`, to which you are changing the process.
`-i` *idtype idlist*	Uses a combination of *xidtype* and *xidlist* to identify the process or processes. The *xidtype* specifies the type of ID, such as the process ID or user ID. Use *xidlist* to identify a list of process IDs or user IDs.

Note – You must be the root user or working in a real-time shell to change a process from, or to, a real-time process. If, as superuser, you change a user process to the real-time class, the user cannot subsequently change the real-time scheduling parameters by using the `priocntl -s` command.

3 Verify the process status.

```
# ps -ecl | grep idlist
```

Example 10–8 Changing the Class of a Process (`priocntl`)

The following example shows how to change all the processes that belong to user `15249` to real-time processes.

```
# priocntl -s -c RT -i uid 15249
# ps -ecl | grep 15249
```

Changing the Priority of a Timesharing Process (`nice`)

The `nice` command is only supported for backward compatibility to previous releases. The `priocntl` command provides more flexibility in managing processes.

The priority of a process is determined by the policies of its scheduling class and by its *nice number*. Each timesharing process has a global priority. The global priority is calculated by adding the user-supplied priority, which can be influenced by the `nice` or `priocntl` commands, and the system-calculated priority.

The execution priority number of a process is assigned by the operating system. The priority number is determined by several factors, including the process's scheduling class, how much CPU time it has used, and in the case of a timesharing process, its `nice` number.

Each timesharing process starts with a default `nice` number, which it inherits from its parent process. The `nice` number is shown in the `NI` column of the `ps` report.

A user can lower the priority of a process by increasing its user-supplied priority. However, only superuser can lower a `nice` number to increase the priority of a process. This restriction prevents users from increasing the priorities of their own processes, thereby monopolizing a greater share of the CPU.

The `nice` numbers range from 0 to +39, with 0 representing the highest priority. The default `nice` value for each timesharing process is 20. Two versions of the command are available: the standard version, `/usr/bin/nice`, and the C shell built-in command.

How to Change the Priority of a Process (`nice`)

Using this procedure, a user can lower the priority of a process. However, the `root` user can raise or lower the priority of a process.

1. **Determine whether you want to change the priority of a process, either as a user or as superuser. Then, select one of the following:**

   - **As a user, follow the examples in Step 2 to lower the priority of a command.**
   - **As a superuser, follow the examples in Step 3 to raise or lower priorities of a command.**

2. **As a user, lower the priority of a command by increasing the `nice` number.**

   The following `nice` command executes *command-name* with a lower priority by raising the nice number by 5 units.

   ```
   $ /usr/bin/nice -5 command-name
   ```

   In the preceding command, the minus sign designates that what follows is an option. This command could also be specified as follows:

   ```
   $ /usr/bin/nice -n 5 command-name
   ```

   The following `nice` command lowers the priority of *command-name* by raising the nice number by the default increment of 10 units, but not beyond the maximum value of 39.

   ```
   $ /usr/bin/nice command-name
   ```

3 As superuser, raise or lower the priority of a command by changing the `nice` number.

The following `nice` command raises the priority of *command-name* by lowering the nice number by 10 units, but not below the minimum value of 0.

```
# /usr/bin/nice --10 command-name
```

In the preceding command, the first minus sign designates that what follows is an option. The second minus sign indicates a negative number.

The following `nice` command lowers the priority of *command-name* by raising the nice number by 5 units, but not beyond the maximum value of 39.

```
# /usr/bin/nice -5 command-name
```

See Also For more information, see the `nice(1)` man page.

Troubleshooting Problems With System Processes

Here are some tips on obvious problems you might encounter:

- Look for several identical jobs that are owned by the same user. This problem might occur because of a running script that starts a lot of background jobs without waiting for any of the jobs to finish.
- Look for a process that has accumulated a large amount of CPU time. You can identify this problem by checking the `TIME` field in the `ps` output. Possibly, the process is in an endless loop.
- Look for a process that is running with a priority that is too high. Use the `ps -c` command to check the `CLS` field, which displays the scheduling class of each process. A process executing as a real-time (`RT`) process can monopolize the CPU. Or, look for a timesharing (`TS`) process with a high `nice` number. A user with superuser privileges might have increased the priority of a process. The system administrator can lower the priority by using the `nice` command.
- Look for a runaway process. A runaway process progressively uses more and more CPU time. You can identify this problem by looking at the time when the process started (`STIME`) and by watching the cumulation of CPU time (`TIME`) for a while.

◆ ◆ ◆ CHAPTER 11

Monitoring System Performance (Tasks)

Achieving good performance from a computer or network is an important part of system administration. This chapter provides overview of some factors that contribute to managing the performance of the computer systems in your care. In addition, this chapter describes procedures for monitoring system performance by using the `vmstat`, `iostat`, `df`, and `sar` commands.

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

- "Where to Find System Performance Tasks" on page 181
- "System Performance and System Resources" on page 182
- "Processes and System Performance" on page 182
- "About Monitoring System Performance" on page 184
- "Displaying System Performance Information (Task Map)" on page 185
- "Monitoring System Activities (Task Map)" on page 193

Where to Find System Performance Tasks

System Performance Task	For More Information
Manage processes	Chapter 10, "Managing System Processes (Tasks)"
Monitor system performance	Chapter 11, "Monitoring System Performance (Tasks)"
Change tunable parameters	*Oracle Solaris Tunable Parameters Reference Manual*
Manage system performance tasks	Chapter 2, "Projects and Tasks (Overview)," in *Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management*

System Performance Task	For More Information
Manage processes with FX and FS schedulers	Chapter 8, "Fair Share Scheduler (Overview)," in *Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management*

System Performance and System Resources

The performance of a computer system depends upon how the system uses and allocates its resources. Monitor your system's performance regularly so that you know how it behaves under normal conditions. You should have a good idea of what to expect, and be able to recognize a problem when it occurs.

System resources that affect performance are described in the following table.

System Resource	Description
Central processing unit (CPU)	The CPU processes instructions by fetching instructions from memory and executing them.
Input/output (I/O) devices	I/O devices transfer information into and out of the computer. Such a device could be a terminal and keyboard, a disk drive, or a printer.
Memory	Physical (or main) memory is the amount of random access memory (RAM) on the system.

Chapter 11, "Monitoring System Performance (Tasks)," describes the tools that display statistics about the system's activity and performance.

Processes and System Performance

The following table describes terms that are related to processes.

TABLE 11–1 Process Terminology

Term	Description
Process	Any system activity or job. Each time you boot a system, execute a command, or start an application, the system activates one or more processes.

TABLE 11–1 Process Terminology *(Continued)*

Term	Description
Lightweight process (LWP)	A virtual CPU or execution resource. LWPs are scheduled by the kernel to use available CPU resources based on their scheduling class and priority. LWPs include a kernel thread and an LWP. A kernel thread contains information that has to be in memory all the time. An LWP contains information that is swappable.
Application thread	A series of instructions with a separate stack that can execute independently in a user's address space. Application threads can be multiplexed on top of LWPs.

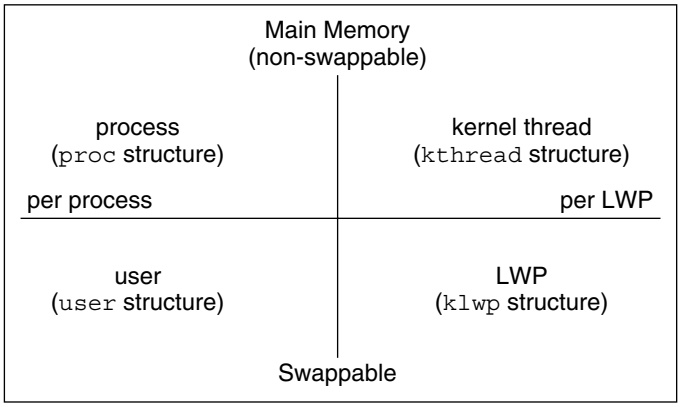
A process can consist of multiple LWPs and multiple application threads. The kernel schedules a kernel-thread structure, which is the scheduling entity in the SunOS environment. Various process structures are described in the following table.

TABLE 11–2 Process Structures

Structure	Description
`proc`	Contains information that pertains to the whole process and must be in main memory all the time
`kthread`	Contains information that pertains to one LWP and must be in main memory all the time
`user`	Contains the "per process" information that is swappable
`klwp`	Contains the "per LWP process" information that is swappable

The following figure illustrates the relationships among these process structures.

FIGURE 11–1 Relationships Among Process Structures



Most process resources are accessible to all the threads in the process. Almost all process virtual memory is shared. A change in shared data by one thread is available to the other threads in the process.

About Monitoring System Performance

While your computer is running, counters in the operating system are incremented to track various system activities.

System activities that are tracked are as follows:

- Central processing unit (CPU) utilization
- Buffer usage
- Disk and tape input/output (I/O) activity
- Terminal device activity
- System call activity
- Context switching
- File access
- Queue activity
- Kernel tables
- Interprocess communication
- Paging
- Free memory and swap space
- Kernel memory allocation (KMA)

Monitoring Tools

The Oracle Solaris software provides several tools to help you track how your system is performing.

TABLE 11–3 Performance Monitoring Tools

Command	Description	For More Information
`cpustat` and `cputrack` commands	Monitors performance of a system or a process using CPU performance counters.	`cpustat`(1M) and `cputrack`(1)
`netstat` and `nfsstat` commands	Displays information about network performance.	`netstat`(1M) and `nfsstat`(1M)
`ps` and `prstat` commands	Displays information about active processes.	Chapter 10, "Managing System Processes (Tasks)"
`sar` and `sadc` commands	Collects and reports on system activity data.	Chapter 11, "Monitoring System Performance (Tasks)"
`swap` command	Displays information about available swap space on your system.	Chapter 19, "Configuring Additional Swap Space (Tasks)," in *Oracle Solaris Administration: Devices and File Systems*
`vmstat` and `iostat` commands	Summarizes system activity data, such as virtual memory statistics, disk usage, and CPU activity.	Chapter 11, "Monitoring System Performance (Tasks)"
`cputrack` and `cpustat` commands	Assists in accessing hardware performance counter facilities provided by microprocessors.	`cputrack`(1) and `cpustat`(1M) man pages
`kstat` and `mpstat` commands	Examines the available kernel statistics, or `kstats`, on the system and reports those statistics which match the criteria specified on the command line. The `mpstat` command reports processor statistics in tabular form.	`kstat`(1M) and `mpstat`(1M) man pages.

Displaying System Performance Information (Task Map)

Task	Description	For Instructions
Display virtual memory Statistics.	Collect virtual memory statistics by using the `vmstat` command.	"How to Display Virtual Memory Statistics (`vmstat`)" on page 187
Display system event information.	Display system event information by using the `vmstat` command with the `-s` option.	"How to Display System Event Information (`vmstat -s`)" on page 188

Task	Description	For Instructions
Display swapping statistics.	Use the vmstat command with the -S option to display swapping statistics.	"How to Display Swapping Statistics (vmstat -S)" on page 188
Display interrupts per device.	Use the vmstat command with the -i option to show the number of interrupts per device.	"How to Display Interrupts Per Device (vmstat -i)" on page 189
Display disk utilization.	Use the iostat command to report disk input and output statistics.	"How to Display Disk Utilization Information (iostat)" on page 190
Display extended disk statistics.	Use the iostat command with the -xtcoption to display extended disk statistics.	"How to Display Extended Disk Statistics (iostat -xtc)" on page 191
Display disk space information.	The df -k command displays disk space information in Kbytes.	"How to Display Disk Space Information (df -k)" on page 192

Displaying Virtual Memory Statistics (vmstat)

You can use the vmstat command to report virtual memory statistics and information about system events such as CPU load, paging, number of context switches, device interrupts, and system calls. The vmstat command can also display statistics on swapping, cache flushing, and interrupts.

TABLE 11–4 Output From the vmstat Command

Category	Field Name	Description
procs		Reports on the following:
	r	The number of kernel threads in the dispatch queue
	b	The number of blocked kernel threads that are waiting for resources
	w	The number of swapped out LWPs that are waiting for processing resources to finish
memory		Reports on usage of real memory and virtual memory:
	swap	Available swap space
	free	Size of the free list
page		Reports on page faults and paging activity, in units per second:
	re	Pages reclaimed
	mf	Minor faults and major faults

TABLE 11–4 Output From the vmstat Command *(Continued)*

Category	Field Name	Description
	pi	Kbytes paged in
	po	Kbytes paged out
	fr	Kbytes freed
	de	Anticipated memory that is needed by recently swapped-in processes
	sr	Pages scanned by the page daemon not currently in use. If sr does not equal zero, the page daemon has been running.
disk		Reports the number of disk operations per second, showing data on up to four disks
faults		Reports the trap/interrupt rates per second:
	in	Interrupts per second
	sy	System calls per second
	cs	CPU context switch rate
cpu		Reports on the use of CPU time:
	us	User time
	sy	System time
	id	Idle time

For a more detailed description of this command, see the vmstat(1M) man page.

▼ How to Display Virtual Memory Statistics (vmstat)

- **Collect virtual memory statistics by using the vmstat command with a time interval in seconds.**

```
$ vmstat n
```

where *n* is the interval in seconds between reports.

Example 11–1 Displaying Virtual Memory Statistics

The following example shows the vmstat display of statistics that were gathered at five-second intervals:

```
$ vmstat 5
kthr      memory            page            disk          faults      cpu
 r b w   swap  free  re  mf pi po fr de sr dd f0 s1 --   in   sy   cs us sy id
```

```
 0 0 0 863160 365680  0   3  1  0  0  0  0  0  0  0  406  378  209  1  0 99
 0 0 0 765640 208568  0  36  0  0  0  0  0  0  0  0  479 4445 1378  3  3 94
 0 0 0 765640 208568  0   0  0  0  0  0  0  0  0  0  423  214  235  0  0 100
 0 0 0 765712 208640  0   0  0  0  0  0  0  3  0  0  412  158  181  0  0 100
 0 0 0 765832 208760  0   0  0  0  0  0  0  0  0  0  402  157  179  0  0 100
 0 0 0 765832 208760  0   0  0  0  0  0  0  0  0  0  403  153  182  0  0 100
 0 0 0 765832 208760  0   0  0  0  0  0  0  0  0  0  402  168  177  0  0 100
 0 0 0 765832 208760  0   0  0  0  0  0  0  0  0  0  402  153  178  0  0 100
 0 0 0 765832 208760  0  18  0  0  0  0  0  0  0  0  407  165  186  0  0 100
```

▼ How to Display System Event Information (vmstat -s)

- **Run the vmstat -s command to show how many system events have taken place since the last time the system was booted.**

```
$ vmstat -s
        0 swap ins
        0 swap outs
        0 pages swapped in
        0 pages swapped out
   522586 total address trans. faults taken
    17006 page ins
       25 page outs
    23361 pages paged in
       28 pages paged out
    45594 total reclaims
    45592 reclaims from free list
        0 micro (hat) faults
   522586 minor (as) faults
    16189 major faults
    98241 copy-on-write faults
   137280 zero fill page faults
    45052 pages examined by the clock daemon
        0 revolutions of the clock hand
       26 pages freed by the clock daemon
     2857 forks
       78 vforks
     1647 execs
 34673885 cpu context switches
 65943468 device interrupts
   711250 traps
 63957605 system calls
  3523925 total name lookups (cache hits 99%)
    92590 user   cpu
    65952 system cpu
 16085832 idle   cpu
     7450 wait   cpu
```

▼ How to Display Swapping Statistics (vmstat -S)

- **Run vmstat -S to show swapping statistics.**

```
$ vmstat -S
kthr      memory            page            disk          faults      cpu
```

```
 r b w   swap  free  si  so pi po fr de sr dd f0 s1 --   in   sy   cs us sy id
 0 0 0 862608 364792  0   0  1  0  0  0  0  0  0  0  0  406  394  213  1  0 99
```

The swapping statistics fields are described in the following list. For a description of the other fields, see Table 11–4.

`si` Average number of LWPs that are swapped in per second

`so` Number of whole processes that are swapped out

Note – The `vmstat` command truncates the output of `si` and `so` fields. Use the `sar` command to display a more accurate accounting of swap statistics.

▼ How to Display Interrupts Per Device (`vmstat -i`)

- **Run the `vmstat -i` command to show the number of interrupts per device.**

Example 11–2 Displaying Interrupts Per Device

The following example shows output from the `vmstat -i` command.

```
$ vmstat -i
interrupt         total     rate
--------------------------------
clock          52163269      100
esp0            2600077        4
zsc0              25341        0
zsc1              48917        0
cgsixc0             459        0
lec0             400882        0
fdc0                 14        0
bppc0                 0        0
audiocs0              0        0
--------------------------------
Total          55238959      105
```

Displaying Disk Utilization Information (`iostat`)

Use the `iostat` command to report statistics about disk input and output, and to produce measures of throughput, utilization, queue lengths, transaction rates, and service time. For a detailed description of this command, refer to the `iostat(1M)` man page.

▼ How to Display Disk Utilization Information (iostat)

- **You can display disk utilization information by using the `iostat` command with a time interval in seconds.**

```
$ iostat 5
     tty          fd0           sd3          nfs1         nfs31          cpu
tin tout kps tps serv  kps tps serv  kps tps serv  kps tps serv  us sy wt id
  0    1   0   0  410    3   0   29    0   0    9    3   0   47   4  2  0 94
```

The first line of output shows the statistics since the last time the system was booted. Each subsequent line shows the interval statistics. The default is to show statistics for the terminal (`tty`), disks (`fd` and `sd`), and CPU (`cpu`).

Example 11–3 Displaying Disk Utilization Information

The following example shows disk statistics that were gathered every five seconds.

```
$ iostat 5
tty         sd0           sd6           nfs1         nfs49          cpu
tin tout kps tps serv  kps tps serv  kps tps serv  kps tps serv  us sy wt id
 0    0   1   0   49    0   0    0    0   0    0    0   0   15   0  0  0 100
 0   47   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16  44   6  132    0   0    0    0   0    0    0   0    0   0  0  1 99
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   3   1   23    0   0    0    0   0    0    0   0    0   0  0  1 99
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
```

The following table describes the fields in the output of the `iostat` *n* command.

Device Type	Field Name	Description
Terminal	Device Type	
	`tin`	Number of characters in the terminal input queue
	`tout`	Number of characters in the terminal output queue
Disk	Device Type	
	`bps`	Blocks per second

Device Type	Field Name	Description
	tps	Transactions per second
	serv	Average service time, in milliseconds
CPU	Device Type	
	us	In user mode
	sy	In system mode
	wt	Waiting for I/O
	id	Idle

How to Display Extended Disk Statistics (iostat -xtc)

- **Run the iostat -xtc command to display extended disk statistics.**

```
$ iostat -xtc
                 extended device statistics                      tty         cpu
device       r/s    w/s   kr/s   kw/s wait actv  svc_t  %w  %b  tin tout  us sy wt id
fd0          0.0    0.0    0.0    0.0  0.0  0.0    0.0   0   0    0    0   0  0  0 100
sd0          0.0    0.0    0.4    0.4  0.0  0.0   49.5   0   0
sd6          0.0    0.0    0.0    0.0  0.0  0.0    0.0   0   0
nfs1         0.0    0.0    0.0    0.0  0.0  0.0    0.0   0   0
nfs49        0.0    0.0    0.0    0.0  0.0  0.0   15.1   0   0
nfs53        0.0    0.0    0.4    0.0  0.0  0.0   24.5   0   0
nfs54        0.0    0.0    0.0    0.0  0.0  0.0    6.3   0   0
nfs55        0.0    0.0    0.0    0.0  0.0  0.0    4.9   0   0
```

The iostat -xtc command displays a line of output for each disk. The output fields are described in the following list.

r/s — Reads per second

w/s — Writes per second

kr/s — Kbytes read per second

kw/s — Kbytes written per second

wait — Average number of transactions that are waiting for service (queue length)

actv — Average number of transactions that are actively being serviced

svc_t — Average service time, in milliseconds

%w — Percentage of time that the queue is not empty

%b — Percentage of time that the disk is busy

Displaying Disk Space Statistics (df)

Use the df command to show the amount of free disk space on each mounted disk. The *usable* disk space that is reported by df reflects only 90 percent of full capacity, as the reporting statistics allows for 10 percent above the total available space. This *head room* normally stays empty for better performance.

The percentage of disk space actually reported by the df command is used space divided by usable space.

If the file system exceeds 90 percent capacity, you could transfer files to a disk that is not as full by using the cp command. Alternately, you could transfer files to a tape by using the tar or cpio commands. Or, you could remove the files.

For a detailed description of this command, see the df(1M) man page.

▼ How to Display Disk Space Information (df -k)

- **Use the df -k command to display disk space information in Kbytes.**

```
$ df -k
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t3d0s0     192807   40231  133296    24%    /
```

Example 11–4 Displaying File System Information

The following example shows the output from the df -k command.

```
$ df -k
Filesystem           1024-blocks        Used   Available Capacity  Mounted on
rpool/ROOT/solaris-161   191987712     6004395   140577816     5%    /
/devices                         0           0           0     0%    /devices
/dev                             0           0           0     0%    /dev
ctfs                             0           0           0     0%    /system/contract
proc                             0           0           0     0%    /proc
mnttab                           0           0           0     0%    /etc/mnttab
swap                       4184236         496     4183740     1%    /system/volatile
objfs                            0           0           0     0%    /system/object
sharefs                          0           0           0     0%    /etc/dfs/sharetab
/usr/lib/libc/libc_hwcap1.so.1   146582211     6004395   140577816     5%    /lib/libc.so.1
fd                               0           0           0     0%    /dev/fd
swap                       4183784          60     4183724     1%    /tmp
rpool/export             191987712          35   140577816     1%    /export
rpool/export/home        191987712          32   140577816     1%    /export/home
rpool/export/home/123    191987712    13108813   140577816     9%    /export/home/123
rpool/export/repo        191987712    11187204   140577816     8%    /export/repo
rpool/export/repo2010_11   191987712          31   140577816     1%    /export/repo2010_11
rpool                    191987712     5238974   140577816     4%    /rpool
/export/home/123         153686630    13108813   140577816     9%    /home/123
```

The following table describes the output of the `df -k` command.

Field Name	Description
`kbytes`	Total size of usable space in the file system
`used`	Amount of space used
`avail`	Amount of space available for use
`capacity`	Amount of space used, as a percentage of the total capacity
`mounted on`	Mount point

Example 11–5 Displaying File System Information by Using the `df` Command Without Any Options

When the `df` command is used without operands or options, it reports on all mounted file systems, as shown in the following example:

```
$ df
/                  (rpool/ROOT/solaris):100715496 blocks 100715496 files
/devices           (/devices           ):       0 blocks        0 files
/dev               (/dev               ):       0 blocks        0 files
/system/contract   (ctfs               ):       0 blocks 2147483601 files
/proc              (proc               ):       0 blocks    29946 files
/etc/mnttab        (mnttab             ):       0 blocks        0 files
/system/volatile   (swap               ):42257568 blocks  2276112 files
/system/object     (objfs              ):       0 blocks 2147483441 files
/etc/dfs/sharetab  (sharefs            ):       0 blocks 2147483646 files
/dev/fd            (fd                 ):       0 blocks        0 files
/tmp               (swap               ):42257568 blocks  2276112 files
/export            (rpool/export       ):100715496 blocks 100715496 files
/export/home       (rpool/export/home  ):100715496 blocks 100715496 files
/export/home/admin (rpool/export/home/admin):100715496 blocks 100715496 files
/rpool             (rpool              ):100715496 blocks 100715496 files
/export/repo2010_11(rpool/export/repo2010_11):281155639 blocks 281155639 files
/rpool             (rpool              ):281155639 blocks 281155639 files
```

Monitoring System Activities (Task Map)

Task	Description	For Instructions
Check file access.	Display file access operation status by using the `sar` command with the `-a` option.	"How to Check File Access (`sar -a`)" on page 195
Check buffer activity.	Display buffer activity statistics by using the `sar` command with the `-b` option.	"How to Check Buffer Activity (`sar -b`)" on page 196

Task	Description	For Instructions
Check system call statistics.	Display system call statistics by using the `sar` command with the `-c` option.	"How to Check System Call Statistics (`sar -c`)" on page 197
Check disk activity.	Check disk activity by using the `sar` command with the `-d` option.	"How to Check Disk Activity (`sar -d`)" on page 199
Check page-out and memory.	Use the `sar` command with the `-g` option to display page-out memory freeing activities.	"How to Check Page-Out and Memory (`sar -g`)" on page 200
Check kernel memory allocation.	The kernel memory allocation (KMA) allows a kernel subsystem to allocate and free memory, as needed. Use the `sar` command with the `-k` option to check KMA.	"How to Check Kernel Memory Allocation (`sar -k`)" on page 202
Check interprocess communication.	Use the `sar` command with the `-m` option to report interprocess communication activities.	"How to Check Interprocess Communication (`sar -m`)" on page 203
Check page-in activity.	Use the `sar` command with the `-p` option to report page-in activity.	"How to Check Page-In Activity (`sar -p`)" on page 204
Check queue activity.	Use the `sar` command with the `-q` option to check the following: ■ Average queue length while queue is occupied ■ Percentage of time that the queue is occupied	"How to Check Queue Activity (`sar -q`)" on page 205
Check unused memory.	Use the `sar` command with the `-r` option to report the number of memory pages and swap file disk blocks that are currently used.	"How to Check Unused Memory (`sar -r`)" on page 206
Check CPU utilization.	Use the `sar` command with the `-u` option to display CPU utilization statistics.	"How to Check CPU Utilization (`sar -u`)" on page 207
Check system table status.	Use the `sar` command with the `-v` option to report status on the following system tables: ■ Process ■ Inode ■ File ■ Shared memory record	"How to Check System Table Status (`sar -v`)" on page 208
Check swapping activity.	Use the `sar` command with the `-w` option to check swapping activity.	"How to Check Swapping Activity (`sar -w`)" on page 209
Check terminal activity.	Use the `sar` command with the `-y` option to monitor terminal device activity.	"How to Check Terminal Activity (`sar -y`)" on page 210
Check overall system performance.	The `sar -A` command displays statistics from all options to provide overall system performance information.	"How to Check Overall System Performance (`sar -A`)" on page 211

Task	Description	For Instructions
Set up automatic data collection.	To set up your system to collect data automatically and to run the sar commands, do the following: ■ Run the svcadm enable system/sar:default command ■ Edit the /var/spool/cron/crontabs/sys file	"How to Set Up Automatic Data Collection" on page 214

Monitoring System Activities (sar)

Use the sar command to perform the following tasks:

- Organize and view data about system activity.
- Access system activity data on a special request basis.
- Generate automatic reports to measure and monitor system performance, as well as special request reports to pinpoint specific performance problems. For information about how to set up the sar command to run on your system, as well as a description of these tools, see "Collecting System Activity Data Automatically (sar)" on page 211.

For a detailed description of this command, see the sar(1) man page.

▼ How to Check File Access (sar -a)

- **Display file access operation statistics with the sar -a command.**

```
$ sar -a

SunOS t2k-brm-24 5.10 Generic_144500-10 sun4v    ...

00:00:00  iget/s namei/s dirbk/s
01:00:00       0       3       0
02:00:00       0       3       0
03:00:00       0       3       0
04:00:00       0       3       0
05:00:00       0       3       0
06:00:00       0       3       0
07:00:00       0       3       0
08:00:00       0       3       0
08:20:01       0       3       0
08:40:00       0       3       0
09:00:00       0       3       0
09:20:01       0      10       0
09:40:01       0       1       0
10:00:02       0       5       0

Average        0       4       0
```

The following list describes the field names and description of operating system routines that are reported by the sar -a command.

iget/s — The number of requests made for inodes that were not in the directory name look-up cache (DNLC).

namei/s — The number of file system path searches per second. If namei does not find a directory name in the DNLC, it calls iget to get the inode for either a file or directory. Hence, most igets are the result of DNLC misses.

dirbk/s — The number of directory block reads issued per second.

The larger the reported values for these operating system routines, the more time the kernel is spending to access user files. The amount of time reflects how heavily programs and applications are using the file systems. The -a option is helpful for viewing how disk-dependent an application is.

▼ How to Check Buffer Activity (sar -b)

- **Display buffer activity statistics with the sar -b command.**

  The buffer is used to cache metadata. Metadata includes inodes, cylinder group blocks, and indirect blocks.

```
$ sar -b
00:00:00 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
01:00:00       0       0     100       0       0      55       0       0
```

Example 11–6 Checking Buffer Activity (sar -b)

The following example of sar -b command output shows that the %rcache and %wcache buffers are not causing any slowdowns. All the data is within acceptable limits.

```
$ sar -b

SunOS t2k-brm-24 5.10 Generic_144500-10 sun4v    ...

00:00:04 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
01:00:00       0       0     100       0       0      94       0       0
02:00:01       0       0     100       0       0      94       0       0
03:00:00       0       0     100       0       0      92       0       0
04:00:00       0       1     100       0       1      94       0       0
05:00:00       0       0     100       0       0      93       0       0
06:00:00       0       0     100       0       0      93       0       0
07:00:00       0       0     100       0       0      93       0       0
08:00:00       0       0     100       0       0      93       0       0
08:20:00       0       1     100       0       1      94       0       0
08:40:01       0       1     100       0       1      93       0       0
09:00:00       0       1     100       0       1      93       0       0
09:20:00       0       1     100       0       1      93       0       0
```

```
09:40:00       0       2     100       0       1      89       0       0
10:00:00       0       9     100       0       5      92       0       0
10:20:00       0       0     100       0       0      68       0       0
10:40:00       0       1      98       0       1      70       0       0
11:00:00       0       1     100       0       1      75       0       0

Average        0       1     100       0       1      91       0       0
```

The following table describes the buffer activities that are displayed by the `-b` option.

Field Name	Description
`bread/s`	Average number of reads per second that are submitted to the buffer cache from the disk
`lread/s`	Average number of logical reads per second from the buffer cache
`%rcache`	Fraction of logical reads that are found in the buffer cache (100 % minus the ratio of `bread/s` to `lread/s`)
`bwrit/s`	Average number of physical blocks (512 bytes) that are written from the buffer cache to disk, per second
`lwrit/s`	Average number of logical writes to the buffer cache, per second
`%wcache`	Fraction of logical writes that are found in the buffer cache (100 % minus the ratio of `bwrit/s` to `lwrit/s`)
`pread/s`	Average number of physical reads, per second, that use character device interfaces
`pwrit/s`	Average number of physical write requests, per second, that use character device interfaces

The most important entries are the cache hit ratios `%rcache` and `%wcache`. These entries measure the effectiveness of system buffering. If `%rcache` falls below 90 percent, or if `%wcache` falls below 65 percent, it might be possible to improve performance by increasing the buffer space.

▼ How to Check System Call Statistics (`sar -c`)

- **Display system call statistics by using the `sar -c` command.**

```
$ sar -c
00:00:00 scall/s sread/s swrit/s  fork/s  exec/s rchar/s wchar/s
01:00:00      38       2       2    0.00    0.00     149     120
```

Example 11–7 Checking System Call Statistics (`sar -c`)

The following example shows output from the `sar -c` command.

```
$ sar -c

SunOS balmy 5.10 Generic_144500-10 sun4v ...
00:00:04 scall/s sread/s swrit/s  fork/s  exec/s rchar/s wchar/s
01:00:00      89      14       9    0.01    0.00    2906    2394
02:00:01      89      14       9    0.01    0.00    2905    2393
03:00:00      89      14       9    0.01    0.00    2908    2393
04:00:00      90      14       9    0.01    0.00    2912    2393
05:00:00      89      14       9    0.01    0.00    2905    2393
06:00:00      89      14       9    0.01    0.00    2905    2393
07:00:00      89      14       9    0.01    0.00    2905    2393
08:00:00      89      14       9    0.01    0.00    2906    2393
08:20:00      90      14       9    0.01    0.01    2914    2395
08:40:01      90      14       9    0.01    0.00    2914    2396
09:00:00      90      14       9    0.01    0.01    2915    2396
09:20:00      90      14       9    0.01    0.01    2915    2396
09:40:00     880     207     156    0.08    0.08   26671    9290
10:00:00    2020     530     322    0.14    0.13   57675   36393
10:20:00     853     129      75    0.02    0.01   10500    8594
10:40:00    2061     524     450    0.08    0.08  579217  567072
11:00:00    1658     404     350    0.07    0.06 1152916 1144203

Average      302      66      49    0.02    0.01   57842   55544
```

The following table describes the system call categories that are reported by the `-c` option. Typically, reads and writes account for about half of the total system calls. However, the percentage varies greatly with the activities that are being performed by the system.

Field Name	Description
`scall/s`	The number of all types of system calls per second, which is generally about 30 per second on a system with 4 to 6 users.
`sread/s`	The number of `read` system calls per second.
`swrit/s`	The number of `write` system calls per second.
`fork/s`	The number of `fork` system calls per second, which is about 0.5 per second on a system with 4 to 6 users. This number increases if shell scripts are running.
`exec/s`	The number of `exec` system calls per second. If `exec/s` divided by `fork/s` is greater than 3, look for inefficient `PATH` variables.
`rchar/s`	The number of characters (bytes) transferred by `read` system calls per second.
`wchar/s`	The number of characters (bytes) transferred by `write` system calls per second.

▼ How to Check Disk Activity (sar -d)

- **Display disk activity statistics with the `sar -d` command.**

```
$ sar -d

00:00:00   device        %busy   avque   r+w/s  blks/s  avwait  avserv
```

Example 11–8 Checking Disk Activity

This abbreviated example illustrates the output from the `sar -d` command.

```
$ sar -d

SunOS balmy 5.10 Generic_144500-10 sun4v    ...

12:36:32   device        %busy   avque   r+w/s  blks/s  avwait  avserv

12:40:01   dad1             15     0.7      26     399    18.1    10.0
           dad1,a           15     0.7      26     398    18.1    10.0
           dad1,b            0     0.0       0       1     1.0     3.0
           dad1,c            0     0.0       0       0     0.0     0.0
           dad1,h            0     0.0       0       0     0.0     6.0
           fd0               0     0.0       0       0     0.0     0.0
           nfs1              0     0.0       0       0     0.0     0.0
           nfs2              1     0.0       1      12     0.0    13.2
           nfs3              0     0.0       0       2     0.0     1.9
           nfs4              0     0.0       0       0     0.0     7.0
           nfs5              0     0.0       0       0     0.0    57.1
           nfs6              1     0.0       6     125     4.3     3.2
           nfs7              0     0.0       0       0     0.0     6.0
           sd1               0     0.0       0       0     0.0     5.4
           ohci0,bu          0     0.0       0       0     0.0     0.0
           ohci0,ct          0     0.0       0       0     0.0     0.0
           ohci0,in          0     0.0       7       0     0.0     0.0
           ohci0,is          0     0.0       0       0     0.0     0.0
           ohci0,to          0     0.0       7       0     0.0     0.0
```

The following table describes the disk device activities that are reported by the `-d` option.

Field Name	Description
`device`	Name of the disk device that is being monitored.
`%busy`	Portion of time the device was busy servicing a transfer request.
`avque`	Average number of requests during the time the device was busy servicing a transfer request.
`r+w/s`	Number of read-and-write transfers to the device, per second.
`blks/s`	Number of 512-byte blocks that are transferred to the device, per second.

Field Name	Description
avwait	Average time, in milliseconds, that transfer requests wait idly in the queue. This time is measured only when the queue is occupied.
avserv	Average time, in milliseconds, for a transfer request to be completed by the device. For disks, this value includes seek times, rotational latency times, and data transfer times.

Note that queue lengths and wait times are measured when something is in the queue. If %busy is small, large queues and service times probably represent the periodic efforts by the system to ensure that altered blocks are promptly written to the disk.

▼ How to Check Page-Out and Memory (sar -g)

- **Use the sar -g command to display page-out and memory freeing activities in averages.**

```
$ sar -g
00:00:00  pgout/s ppgout/s pgfree/s pgscan/s %ufs_ipf
01:00:00     0.00     0.00     0.00     0.00     0.00
```

The output displayed by the sar -g command is a good indicator of whether more memory might be needed. Use the ps -elf command to show the number of cycles that are used by the page daemon. A high number of cycles, combined with high values for the pgfree/s and pgscan/s fields, indicates a memory shortage.

The sar -g command also shows whether inodes are being recycled too quickly and causing a loss of reusable pages.

Example 11–9 Checking Page-Out and Memory (sar -g)

The following example shows output from the sar -g command.

```
$ sar -g

SunOS balmy 5.10 Generic_144500-10 sun4v    ...

00:00:00  pgout/s ppgout/s pgfree/s pgscan/s %ufs_ipf
01:00:00     0.00     0.00     0.00     0.00     0.00
02:00:00     0.01     0.01     0.01     0.00     0.00
03:00:00     0.00     0.00     0.00     0.00     0.00
04:00:00     0.00     0.00     0.00     0.00     0.00
05:00:00     0.00     0.00     0.00     0.00     0.00
06:00:00     0.00     0.00     0.00     0.00     0.00
07:00:00     0.00     0.00     0.00     0.00     0.00
08:00:00     0.00     0.00     0.00     0.00     0.00
08:20:01     0.00     0.00     0.00     0.00     0.00
08:40:00     0.00     0.00     0.00     0.00     0.00
09:00:00     0.00     0.00     0.00     0.00     0.00
09:20:01     0.05     0.52     1.62    10.16     0.00
```

```
09:40:01     0.03     0.44     1.47     4.77     0.00
10:00:02     0.13     2.00     4.38    12.28     0.00
10:20:03     0.37     4.68    12.26    33.80     0.00

Average      0.02     0.25     0.64     1.97     0.00
```

The following table describes the output from the `-g` option.

Field Name	Description
`pgout/s`	The number of page-out requests per second.
`ppgout/s`	The actual number of pages that are paged-out, per second. A single page-out request might involve paging-out multiple pages.
`pgfree/s`	The number of pages, per second, that are placed on the free list.
`pgscan/s`	The number of pages, per second, that are scanned by the `page` daemon. If this value is high, the `page` daemon is spending a lot of time checking for free memory. This situation implies that more memory might be needed.
`%ufs_ipf`	The percentage of `ufs` inodes taken off the free list by `iget` that had reusable pages that are associated with them. These pages are flushed and cannot be reclaimed by processes. Thus, this field represents the percentage of `igets` with page flushes. A high value indicates that the free list of inodes is page-bound, and that the number of `ufs` inodes might need to be increased.

Checking Kernel Memory Allocation

The KMA allows a kernel subsystem to allocate and free memory, as needed.

Rather than statically allocating the maximum amount of memory it is expected to require under peak load, the KMA divides requests for memory into three categories:

- Small (less than 256 bytes)
- Large (512 bytes to 4 Kbytes)
- Oversized (greater than 4 Kbytes)

The KMA keeps two pools of memory to satisfy small requests and large requests. The oversized requests are satisfied by allocating memory from the system page allocator.

If you are checking a system that is being used to write drivers or STREAMS that use KMA resources, then the `sar -k` command will likely prove useful. Otherwise, you will probably not need the information it provides. Any driver or module that uses KMA resources, but does not specifically return the resources before it exits, can create a memory leak. A memory leak causes the amount of memory that is allocated by KMA to increase over time. Thus, if the `alloc` fields of the `sar -k` command increase steadily over time, there might be a memory leak. Another

indication of a memory leak is failed requests. If this problem occurs, a memory leak has probably caused KMA to be unable to reserve and allocate memory.

If it appears that a memory leak has occurred, you should check any drivers or STREAMS that might have requested memory from KMA and not returned it.

▼ How to Check Kernel Memory Allocation (sar -k)

- **Use the sar -k command to report on the following activities of the Kernel Memory Allocator (KMA).**

```
$ sar -k
00:00:00 sml_mem   alloc  fail  lg_mem   alloc   fail  ovsz_alloc  fail
01:00:00 2523136 1866512     0 18939904 14762364    0      360448     0
02:00:02 2523136 1861724     0 18939904 14778748    0      360448     0
```

Example 11–10 Checking Kernel Memory Allocation (sar -k)

The following is an abbreviated example of sar -k output.

```
$ sar -k

SunOS balmy 5.10 Generic_144500-10 sun4v    ...
00:00:04 sml_mem   alloc  fail  lg_mem   alloc  fail  ovsz_alloc  fail
01:00:00 6119744 4852865     0 60243968 54334808   156     9666560     0
02:00:01 6119744 4853057     0 60243968 54336088   156     9666560     0
03:00:00 6119744 4853297     0 60243968 54335760   156     9666560     0
04:00:00 6119744 4857673     0 60252160 54375280   156     9666560     0
05:00:00 6119744 4858097     0 60252160 54376240   156     9666560     0
06:00:00 6119744 4858289     0 60252160 54375608   156     9666560     0
07:00:00 6119744 4858793     0 60252160 54442424   156     9666560     0
08:00:00 6119744 4858985     0 60252160 54474552   156     9666560     0
08:20:00 6119744 4858169     0 60252160 54377400   156     9666560     0
08:40:01 6119744 4857345     0 60252160 54376880   156     9666560     0
09:00:00 6119744 4859433     0 60252160 54539752   156     9666560     0
09:20:00 6119744 4858633     0 60252160 54410920   156     9666560     0
09:40:00 6127936 5262064     0 60530688 55619816   156     9666560     0
10:00:00 6545728 5823137     0 62996480 58391136   156     9666560     0
10:20:00 6545728 5758997     0 62996480 57907400   156     9666560     0
10:40:00 6734144 6035759     0 64389120 59743064   156    10493952     0
11:00:00 6996288 6394872     0 65437696 60935936   156    10493952     0

Average  6258044 5150556     0 61138340 55609004   156     9763900     0
```

The following table describes the output from the -k option.

Field Name	Description
sml_mem	The amount of memory, in bytes, that the KMA has available in the small memory request pool. In this pool, here a small request is less than 256 bytes.

Field Name	Description
alloc	The amount of memory, in bytes, that the KMA has allocated from its small memory request pool to small memory requests.
fail	The number of requests for small amounts of memory that failed.
lg_mem	The amount of memory, in bytes, that the KMA has available in the large memory request pool. In this pool, a large request is from 512 bytes to 4 Kbytes.
alloc	The amount of memory, in bytes, that the KMA has allocated from its large memory request pool to large memory requests.
fail	The number of failed requests for large amounts of memory.
ovsz_alloc	The amount of memory that is allocated for oversized requests, which are requests that are greater than 4 Kbytes. These requests are satisfied by the page allocator. Thus, there is no pool.
fail	The number of failed requests for oversized amounts of memory.

How to Check Interprocess Communication (sar -m)

- **Use the sar -m command to report interprocess communication activities.**

```
$ sar -m
00:00:00   msg/s  sema/s
01:00:00    0.00    0.00
```

These figures are usually zero (0.00), unless you are running applications that use messages or semaphores.

The following list describes the output from the -m option.

msg/s The number of message operations (sends and receives) per second

sema/s The number of semaphore operations per second

Example 11–11 Checking Interprocess Communication (sar -m)

The following abbreviated example shows output from the sar -m command.

```
$ sar -m

SunOS balmy 5.10 Generic_144500-10 sun4v    ...

00:00:00   msg/s  sema/s
01:00:00    0.00    0.00
02:00:02    0.00    0.00
03:00:00    0.00    0.00
```

```
04:00:00    0.00    0.00
05:00:01    0.00    0.00
06:00:00    0.00    0.00

Average     0.00    0.00
```

▼ How to Check Page-In Activity (sar -p)

- **Use the sar -p command to report page-in activity, which includes protection and translation faults.**

```
$ sar -p
00:00:00  atch/s  pgin/s ppgin/s  pflt/s  vflt/s slock/s
01:00:00    0.07    0.00    0.00    0.21    0.39    0.00
```

Example 11–12 Checking Page-In Activity (sar -p)

The following example shows output from the sar -p command.

```
$ sar -p

SunOS balmy 5.10 Generic_144500-10 sun4v    ...

00:00:04  atch/s  pgin/s ppgin/s  pflt/s  vflt/s slock/s
01:00:00    0.09    0.00    0.00    0.78    2.02    0.00
02:00:01    0.08    0.00    0.00    0.78    2.02    0.00
03:00:00    0.09    0.00    0.00    0.81    2.07    0.00
04:00:00    0.11    0.01    0.01    0.86    2.18    0.00
05:00:00    0.08    0.00    0.00    0.78    2.02    0.00
06:00:00    0.09    0.00    0.00    0.78    2.02    0.00
07:00:00    0.08    0.00    0.00    0.78    2.02    0.00
08:00:00    0.09    0.00    0.00    0.78    2.02    0.00
08:20:00    0.11    0.00    0.00    0.87    2.24    0.00
08:40:01    0.13    0.00    0.00    0.90    2.29    0.00
09:00:00    0.11    0.00    0.00    0.88    2.24    0.00
09:20:00    0.10    0.00    0.00    0.88    2.24    0.00
09:40:00    2.91    1.80    2.38    4.61   17.62    0.00
10:00:00    2.74    2.03    3.08    8.17   21.76    0.00
10:20:00    0.16    0.04    0.04    1.92    2.96    0.00
10:40:00    2.10    2.50    3.42    6.62   16.51    0.00
11:00:00    3.36    0.87    1.35    3.92   15.12    0.00

Average     0.42    0.22    0.31    1.45    4.00    0.00
```

The following table describes the reported statistics from the -p option.

Field Name	Description
atch/s	The number of page faults, per second, that are satisfied by reclaiming a page currently in memory (attaches per second). Instances include reclaiming an invalid page from the free list and sharing a page of text that is currently being used by another process. An example is two or more processes that are accessing the same program text.
pgin/s	The number of times, per second, that file systems receive page-in requests.
ppgin/s	The number of pages paged in, per second. A single page-in request, such as a soft-lock request (see slock/s) or a large block size, might involve paging-in multiple pages.
pflt/s	The number of page faults from protection errors. Instances of protection faults indicate illegal access to a page and "copy-on-writes." Generally, this number consists primarily of "copy-on-writes."
vflt/s	The number of address translation page faults, per second. These faults are known as validity faults. Validity faults occur when a valid process table entry does not exist for a given virtual address.
slock/s	The number of faults, per second, caused by software lock requests that require physical I/O. An example of the occurrence of a soft-lock request is the transfer of data from a disk to memory. The system locks the page that is to receive the data so that the page cannot be claimed and used by another process.

▼ How to Check Queue Activity (sar -q)

- **Use the sar -q command to report the following information:**
  - The Average queue length while the queue is occupied.
  - The percentage of time that the queue is occupied.

```
$ sar -q
00:00:00 runq-sz %runocc swpq-sz %swpocc
```

The following list describes the output from the -q option.

runq-sz — The number of kernel threads in memory that are waiting for a CPU to run. Typically, this value should be less than 2. Consistently higher values mean that the system might be CPU-bound.

%runocc — The percentage of time that the dispatch queues are occupied.

swpq-sz — The average number of swapped out processes.

`%swpocc` The percentage of time in which the processes are swapped out.

Example 11–13 Checking Queue Activity

The following example shows output from the `sar -q` command. If the `%runocc` value is high (greater than 90 percent) and the `runq-sz` value is greater than 2, the CPU is heavily loaded and response is degraded. In this case, additional CPU capacity might be required to obtain acceptable system response.

```
# sar -q
SunOS balmy 5.10 Generic_144500-10 sun4v    ...

00:00:00 runq-sz %runocc swpq-sz %swpocc
01:00:00     1.0       7     0.0       0
02:00:00     1.0       7     0.0       0
03:00:00     1.0       7     0.0       0
04:00:00     1.0       7     0.0       0
05:00:00     1.0       6     0.0       0
06:00:00     1.0       7     0.0       0

Average      1.0       7     0.0       0
```

▼ How to Check Unused Memory (`sar -r`)

- **Use the `sar -r` command to report the number of memory pages and swap-file disk blocks that are currently unused.**

```
$ sar -r
00:00:00 freemem freeswap
01:00:00    2135   401922
```

The following list describes the output from the `-r` option:

`freemem` The average number of memory pages that are available to user processes over the intervals sampled by the command. Page size is machine-dependent.

`freeswap` The number of 512-byte disk blocks that are available for page swapping.

Example 11–14 Checking Unused Memory (`sar -r`)

The following example shows output from the `sar -r` command.

```
$ sar -r

SunOS balmy 5.10 Generic_144500-10 sun4v    ...

00:00:04 freemem freeswap
01:00:00   44717  1715062
02:00:01   44733  1715496
```

```
03:00:00   44715  1714746
04:00:00   44751  1715403
05:00:00   44784  1714743
06:00:00   44794  1715186
07:00:00   44793  1715159
08:00:00   44786  1714914
08:20:00   44805  1715576
08:40:01   44797  1715347
09:00:00   44761  1713948
09:20:00   44802  1715478
09:40:00   41770  1682239
10:00:00   35401  1610833
10:20:00   34295  1599141
10:40:00   33943  1598425
11:00:00   30500  1561959

Average    43312  1699242
```

▼ How to Check CPU Utilization (`sar -u`)

- **Use the `sar -u` command to display CPU utilization statistics.**

```
$ sar -u
00:00:00    %usr    %sys    %wio   %idle
01:00:00       0       0       0     100
```

The `sar` command without any options is equivalent to the `sar -u` command. At any given moment, the processor is either busy or idle. When busy, the processor is in either user mode or system mode. When idle, the processor is either waiting for I/O completion or "sitting still" with no work to do.

The following list describes output from the `-u` option:

`%usr` Lists the percentage of time that the processor is in user mode.

`%sys` Lists the percentage of time that the processor is in system mode.

`%wio` Lists the percentage of time that the processor is idle and waiting for I/O completion.

`%idle` Lists the percentage of time that the processor is idle and not waiting for I/O.

A high `%wio` value generally means that a disk slowdown has occurred.

Example 11–15 Checking CPU Utilization (`sar -u`)

The following example shows output from the `sar -u` command.

```
$ sar -u

00:00:04    %usr    %sys    %wio   %idle
01:00:00       0       0       0     100
02:00:01       0       0       0     100
```

```
03:00:00       0       0       0     100
04:00:00       0       0       0     100
05:00:00       0       0       0     100
06:00:00       0       0       0     100
07:00:00       0       0       0     100
08:00:00       0       0       0     100
08:20:00       0       0       0      99
08:40:01       0       0       0      99
09:00:00       0       0       0      99
09:20:00       0       0       0      99
09:40:00       4       1       0      95
10:00:00       4       2       0      94
10:20:00       1       1       0      98
10:40:00      18       3       0      79
11:00:00      25       3       0      72

Average        2       0       0      98
```

▼ How to Check System Table Status (`sar -v`)

- **Use the `sar -v` command to report the status of the process table, inode table, file table, and shared memory record table.**

```
$ sar -v
00:00:00  proc-sz    ov  inod-sz    ov  file-sz    ov   lock-sz
01:00:00   43/922     0 2984/4236    0  322/322     0    0/0
```

Example 11–16 Checking System Table Status (`sar -v`)

The following abbreviated example shows output from the `sar -v` command. This example shows that all tables are large enough to have no overflows. These tables are all dynamically allocated based on the amount of physical memory.

```
$ sar -v

00:00:04  proc-sz    ov  inod-sz    ov  file-sz    ov   lock-sz
01:00:00   69/8010    0 3476/34703    0    0/0       0    0/0
02:00:01   69/8010    0 3476/34703    0    0/0       0    0/0
03:00:00   69/8010    0 3476/34703    0    0/0       0    0/0
04:00:00   69/8010    0 3494/34703    0    0/0       0    0/0
05:00:00   69/8010    0 3494/34703    0    0/0       0    0/0
06:00:00   69/8010    0 3494/34703    0    0/0       0    0/0
07:00:00   69/8010    0 3494/34703    0    0/0       0    0/0
08:00:00   69/8010    0 3494/34703    0    0/0       0    0/0
08:20:00   69/8010    0 3494/34703    0    0/0       0    0/0
08:40:01   69/8010    0 3494/34703    0    0/0       0    0/0
09:00:00   69/8010    0 3494/34703    0    0/0       0    0/0
09:20:00   69/8010    0 3494/34703    0    0/0       0    0/0
09:40:00   74/8010    0 3494/34703    0    0/0       0    0/0
10:00:00   75/8010    0 4918/34703    0    0/0       0    0/0
10:20:00   72/8010    0 4918/34703    0    0/0       0    0/0
10:40:00   71/8010    0 5018/34703    0    0/0       0    0/0
```

```
11:00:00   77/8010    0 5018/34703    0    0/0       0    0/0
```

Output from the -v option is described in the following table.

Field Name	Description
proc-sz	The number of process entries (proc structures) that are currently being used, or allocated, in the kernel.
inod-sz	The total number of inodes in memory compared to the maximum number of inodes that are allocated in the kernel. This number is not a strict high watermark. The number can overflow.
file-sz	The size of the open system file table. The sz is given as 0, because space is allocated dynamically for the file table.
ov	The overflows that occur between sampling points for each table.
lock-sz	The number of shared memory record table entries that are currently being used, or allocated, in the kernel. The sz is given as 0 because space is allocated dynamically for the shared memory record table.

How to Check Swapping Activity (sar -w)

- **Use the sar -w command to report swapping and switching activity.**

```
$ sar -w
00:00:00 swpin/s bswin/s swpot/s bswot/s pswch/s
01:00:00    0.00     0.0    0.00     0.0      22
```

The following list describes target values and observations related to the sar -w command output.

swpin/s — The number of LWP transfers into memory per second.

bswin/s — The number of blocks transferred for swap-ins per second. /* (float)PGTOBLK(xx->cvmi.pgswapin) / sec_diff */.

swpot/s — The average number of processes that are swapped out of memory per second. If the number is greater than 1, you might need to increase memory.

bswot/s — The number of blocks that are transferred for swap-outs per second.

pswch/s — The number of kernel thread switches, per second.

Note – All process swap-ins include process initialization.

Example 11–17 Checking Swap Activity (`sar -w`)

The following example shows output from the `sar -w` command.

```
$ sar -w

00:00:04 swpin/s bswin/s swpot/s bswot/s pswch/s
01:00:00    0.00     0.0    0.00     0.0     132
02:00:01    0.00     0.0    0.00     0.0     133
03:00:00    0.00     0.0    0.00     0.0     133
04:00:00    0.00     0.0    0.00     0.0     134
05:00:00    0.00     0.0    0.00     0.0     133
06:00:00    0.00     0.0    0.00     0.0     133
07:00:00    0.00     0.0    0.00     0.0     132
08:00:00    0.00     0.0    0.00     0.0     131
08:20:00    0.00     0.0    0.00     0.0     133
08:40:01    0.00     0.0    0.00     0.0     132
09:00:00    0.00     0.0    0.00     0.0     132
09:20:00    0.00     0.0    0.00     0.0     132
09:40:00    0.00     0.0    0.00     0.0     335
10:00:00    0.00     0.0    0.00     0.0     601
10:20:00    0.00     0.0    0.00     0.0     353
10:40:00    0.00     0.0    0.00     0.0     747
11:00:00    0.00     0.0    0.00     0.0     804

Average     0.00     0.0    0.00     0.0     198
```

▼ How to Check Terminal Activity (`sar -y`)

- **Use the `sar -y` command to monitor terminal device activities.**

```
$ sar -y
00:00:00 rawch/s canch/s outch/s rcvin/s xmtin/s mdmin/s
01:00:00       0       0       0       0       0       0
```

If you have a lot of terminal I/O, you can use this report to determine if any bad lines exist. The activities recorded are defined in the following list.

`rawch/s` Input characters (raw queue) per second.

`canch/s` Input characters that are processed by canon (canonical queue) per second.

`outch/s` Output characters (output queue) per second.

`rcvin/s` Receiver hardware interrupts per second.

`xmtin/s` Transmitter hardware interrupts per second.

`mdmin/s` Modem interrupts per second.

The number of modem interrupts per second (`mdmin/s`) should be close to zero. The receive and transmit interrupts per second (`xmtin/s` and `rcvin/s`) should be less than or equal to the number of incoming or outgoing characters, respectively. If not, check for bad lines.

Example 11–18 Checking Terminal Activity (`sar -y`)

The following example shows output from the `sar -y` command.

```
$ sar -y

00:00:04 rawch/s canch/s outch/s rcvin/s xmtin/s mdmin/s
01:00:00       0       0       0       0       0       0
02:00:01       0       0       0       0       0       0
03:00:00       0       0       0       0       0       0
04:00:00       0       0       0       0       0       0
05:00:00       0       0       0       0       0       0
06:00:00       0       0       0       0       0       0
07:00:00       0       0       0       0       0       0
08:00:00       0       0       0       0       0       0
08:20:00       0       0       0       0       0       0
08:40:01       0       0       0       0       0       0
09:00:00       0       0       0       0       0       0
09:20:00       0       0       0       0       0       0
09:40:00       0       0       1       0       0       0
10:00:00       0       0      37       0       0       0
10:20:00       0       0       0       0       0       0
10:40:00       0       0       3       0       0       0
11:00:00       0       0       3       0       0       0

Average        0       0       1       0       0       0
```

▼ How to Check Overall System Performance (`sar -A`)

- **Use the `sar -A` command to display statistics from all options to provide a view of overall system performance.**

  This command provides a more global perspective. If data from more than a single time segment is shown, the report includes averages.

Collecting System Activity Data Automatically (`sar`)

Three commands are involved in the automatic collection of system activity data: `sadc`, `sa1`, and `sa2`.

The `sadc` data collection utility periodically collects data on system activity and saves the data in a file in binary format, one file for each 24-hour period. You can set up the `sadc` command to run periodically (usually once each hour), and whenever the system boots to multiuser mode. The data files are placed in the `/var/adm/sa` directory. Each file is named `sa`*dd*, where *dd* is the current date. The format of the command is as follows:

`/usr/lib/sa/sadc [`*t n*`] [`*ofile*`]`

The command samples *n* times with an interval of *t* seconds, which should be greater than five seconds between samples. This command then writes to the binary *ofile* file, or to standard output.

Running the sadc Command When Booting

The sadc command should be run at system boot time to record the statistics from when the counters are reset to zero. To make sure that the sadc command is run at boot time, the svcadm enable system/sar:default command writes a record to the daily data file.

The command entry has the following format:

```
/usr/bin/su sys -c "/usr/lib/sa/sadc /var/adm/sa/sa`date +%d`"
```

Running the sadc Command Periodically With the sa1 Script

To generate periodic records, you need to run the sadc command regularly. The simplest way to do so is to uncomment the following lines in the /var/spool/cron/crontabs/sys file:

```
# 0 * * * 0-6 /usr/lib/sa/sa1
# 20,40 8-17 * * 1-5 /usr/lib/sa/sa1
# 5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
```

The sys crontab entries do the following:

- The first two crontab entries cause a record to be written to the /var/adm/sa/sa*dd* file every 20 minutes from 8 a.m. to 5 p.m., Monday through Friday, and every hour on the hour otherwise.
- The third entry writes a record to the /var/adm/sa/sar*dd* file hourly, Monday through Friday, and includes all sar options.

You can change these defaults to meet your needs.

Producing Reports With the sa2 Shell Script

Another shell script, sa2, produces reports rather than binary data files. The sa2 command invokes the sar command and writes the ASCII output to a report file.

Setting Up Automatic Data Collection (sar)

The sar command can be used either to gather system activity data itself or to report what has been collected in the daily activity files that are created by the sadc command.

The sar command has the following formats:

```
sar [-aAbcdgkmpqruvwy] [-o file] t [n]

sar [-aAbcdgkmpqruvwy] [-s time] [-e time] [-i sec] [-f file]
```

The following sar command samples cumulative activity counters in the operating system every *t* seconds, *n* times. The *t* should be five seconds or greater. Otherwise, the command itself might affect the sample. You must specify a time interval in which to take the samples. Otherwise, the command operates according to the second format. The default value of *n* is 1. The following example takes two samples separated by 10 seconds. If the -o option were specified, samples are saved in binary format.

```
$ sar -u 10 2
```

Other important information about the sar command includes the following:

- With no sampling interval or number of samples specified, the sar command extracts data from a previously recorded file. This file is either the file specified by the -f option or, by default, the standard daily activity file, /var/adm/sa/sa*dd*, for the most recent day.
- The -s and -e options define the starting time and the ending time for the report. Starting and ending times are of the form *hh*[:*mm*[:*ss*]], where *hh*, *mm*, and *ss* represent hours, minutes, and seconds.
- The -i option specifies, in seconds, the intervals between record selection. If the -i option is not included, all intervals that are found in the daily activity file are reported.

The following table lists the sar options and their actions.

TABLE 11–5 Options for the sar Command

Option	Actions
-a	Checks file access operations
-b	Checks buffer activity
-c	Checks system calls
-d	Checks activity for each block device
-g	Checks page-out and memory freeing
-k	Checks kernel memory allocation

TABLE 11–5 Options for the sar Command *(Continued)*

Option	Actions
-m	Checks interprocess communication
-nv	Checks system table status
-p	Checks swap and dispatch activity
-q	Checks queue activity
-r	Checks unused memory
-u	Checks CPU utilization
-w	Checks swapping and switching volume
-y	Checks terminal activity
-A	Reports overall system performance, which is the same as entering all options.

Using no option is equivalent to calling the sar command with the -u option.

▼ How to Set Up Automatic Data Collection

1. **Become the root role.**

2. **Run the `svcadm enable system/sar:default` command.**

   This version of the sadc command writes a special record that marks the time when the counters are reset to zero (boot time).

3. **Edit the `/var/spool/cron/crontabs/sys` `crontab` file.**

   Note – Do not edit a crontab file directly. Instead, use the crontab -e command to make changes to an existing crontab file.

   ```
   # crontab -e sys
   ```

4. **Uncomment the following lines:**

   ```
   0 * * * 0-6 /usr/lib/sa/sa1
   20,40 8-17 * * 1-5 /usr/lib/sa/sa1
   5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
   ```

   For more information, see the crontab(1) man page.

◆ ◆ ◆ CHAPTER 12

Managing Software Packages (Tasks)

This chapter describes the most commonly needed commands for managing software that is available as Image Packaging System (IPS) packages.

- "Getting Information About Packages" on page 216
- "Installing and Updating Packages" on page 219

For more information, see the `pkg(1)` man page and *Adding and Updating Oracle Solaris 11 Software Packages*.

Managing Software Packages (Task Map)

TABLE 12–1 Managing Software Packages: Task Map

Task	Description	For Instructions
Determine whether a package is installed and whether an update is available.	Use the `pkg list` command.	"Getting Information About Packages" on page 216
Display information about packages such as name and version.	Use the `pkg info` command.	"Getting Information About Packages" on page 216
Display file system content of packages.	Use the `pkg contents` command.	"Getting Information About Packages" on page 216
Search for packages.	Use the `pkg search` command.	"Getting Information About Packages" on page 216
Install and update packages.	Use the `pkg install` and `pkg update` commands.	"Installing and Updating Packages" on page 219
Update all installed packages.	Use the `pkg update` command.	"Updating All Installed Packages" on page 222

Image Packaging System

Oracle Solaris 11 software is distributed in IPS packages. IPS packages are stored in IPS package repositories, which are populated by IPS publishers. IPS packages are installed into Oracle Solaris 11 images. A subset of the capabilities that are available through the IPS command-line interface is available through the Package Manager graphical user interface.

IPS commands enable you to list, search, install, update, and remove software packages. A single IPS command can update your image to a new operating system release. IPS commands also enable you to manage package publishers and copy or create package repositories.

An *image* is a location where IPS packages are installed and where other IPS operations can be performed.

A *repository* is a location where packages are published and from where packages are retrieved. The location is specified by a Universal Resource Identifier (URI).

A *publisher* identifies a person or organization that publishes one or more packages.

An IPS *package* includes specifications for installable objects such as files, directories, links, drivers, dependencies, groups, users, and license information. Packages also include metadata such as classification, summary, and description. Each IPS package is represented by a Fault Management Resource Identifier (FMRI). The FMRI includes information about the package such as the package name, publisher, version information, and date. When using IPS commands, you can use the smallest portion of the package name that uniquely identifies the package.

Getting Information About Packages

Use the following commands to retrieve information about packages. No special privileges are needed to run any of these commands.

`pkg list`
: The `pkg list` command tells you whether a package is installed in the current image and whether an update is available. With no options or operands, this command lists all packages that are installed in the current image. To narrow your results, provide one or more package names. You can use wildcards in the package names.

`pkg info`
: The `pkg info` command displays information about a package, including the name, installed state, version, packaging date, package size, and the full FMRI. With no options or operands, this command displays information about all packages that are installed in the current image. To narrow your results, provide one or more package names. You can use wildcards in the package names.

`pkg contents`
: The `pkg contents` command displays the file system content of packages. With no options or operands, this command displays path information for all packages that are installed in the current image. Use command options to specify particular package content to display. See the `pkg`(1) man page and *Adding and Updating Oracle Solaris 11 Software Packages* for information about options of the `pkg contents` command. To narrow your results, provide one or more package names. You can use wildcards in the package names.

`pkg search`
: Like the `pkg contents` command, the `pkg search` command examines the contents of packages. While the `pkg contents` command returns the contents, the `pkg search` command returns the names of packages that match the query. By default, `pkg search` query terms are matched exactly and ANDed together. See the `pkg`(1) man page and *Adding and Updating Oracle Solaris 11 Software Packages* for additional options for structuring queries.

The remainder of this section shows some examples of using these commands to display information about packages.

```
$ pkg list amp
pkg list: no packages matching 'amp' installed
```

To list packages that are installed and the newest versions of packages that are not installed but could be installed in this image, use the `-a` option. The "-" in the I column indicates that the package is not installed. The "r" in the O column indicates that the `web/amp` package has been renamed. If you give the command to install the `web/amp` package, the `group/feature/amp` package is installed.

```
$ pkg list -a amp
NAME (PUBLISHER)       VERSION                    IFO
group/feature/amp      0.5.11-0.174.0.0.0.0.2559  ---
web/amp                0.5.11-0.174.0.0.0.0.0     --r
```

An "o" in the O column indicates that package is obsolete. You cannot install a package that is marked obsolete. An "f" in the F column indicates the package is frozen. If a package is frozen, you can only install or update to packages that match the frozen version.

Use the `pkg info` command to display more information. Because you already know the package is not installed in this image, use the `-r` option to query the package repository.

```
$ pkg info -r amp
          Name: group/feature/amp
       Summary: AMP (Apache, MySQL, PHP) Deployment Kit for Oracle Solaris
   Description: Provides a set of components for deployment of an AMP (Apache,
                MySQL, PHP) stack on Oracle Solaris
      Category: Meta Packages/Group Packages (org.opensolaris.category.2008)
                Web Services/Application and Web Servers (org.opensolaris.category.2008)
         State: Not installed
     Publisher: solaris
       Version: 0.5.11
 Build Release: 5.11
```

```
        Branch: 0.174.0.0.0.0.2559
Packaging Date: Wed Sep 21 19:12:55 2011
          Size: 5.45 kB
          FMRI: pkg://solaris/group/feature/amp@0.5.11,5.11-0.174.0.0.0.0.2559:20110921T191255Z

          Name: web/amp
       Summary:
         State: Not installed (Renamed)
    Renamed to: group/feature/amp@0.5.11-0.174.0.0.0.0.0
                consolidation/ips/ips-incorporation
     Publisher: solaris
       Version: 0.5.11
 Build Release: 5.11
        Branch: 0.174.0.0.0.0.0
Packaging Date: Wed Sep 21 19:15:02 2011
          Size: 5.45 kB
          FMRI: pkg://solaris/web/amp@0.5.11,5.11-0.174.0.0.0.0.0:20110921T191502Z
```

To display more information about what is in this package, use the `pkg contents` command.

```
$ pkg contents -r group/feature/amp
pkg: This package delivers no file system content, but may contain metadata. Use
the -o option to specify fields other than 'path', or use the -m option to show
the raw package manifests.
```

The `group/feature/amp` package does not contain information about files because the `group/feature/amp` package is a group package. Instead of files, a group package installs other packages. Use the following options to see which packages are installed by the `group/feature/amp` group package. The `-t depend` option means only list contents that are `depend` actions. The `-a` option specifies that the `type` attribute of the `depend` action must have the value `group`. The `-o fmri` option means output only the package FMRI from each matched `depend` action of type `group`. The `-H` option omits headers.

```
$ pkg contents -rt depend -a type=group -o fmri -H group/feature/amp
database/mysql-51
web/php-52
web/php-52/extension/php-apc
web/php-52/extension/php-mysql
web/server/apache-22
web/server/apache-22/module/apache-dtrace
web/server/apache-22/module/apache-fcgid
web/server/apache-22/module/apache-php5
```

If you know the name of the tool you want to install but do not know the name of the package, use the `search` subcommand. By default, `pkg search` returns the actions that match the query and the package that contains those actions. The following examples show two ways to use `search` to determine that you need to install the package `editor/gnu-emacs` to get the `emacs` tool.

```
$ pkg search /usr/bin/emacs
INDEX    ACTION VALUE                  PACKAGE
path     file   usr/bin/emacs          pkg:/editor/gnu-emacs@23.1-0.173.0.0.0.0.487
$ pkg search file::emacs
```

```
INDEX    ACTION VALUE                  PACKAGE
basename file   usr/bin/emacs          pkg:/editor/gnu-emacs@23.1-0.173.0.0.0.0.487
basename file   usr/share/info/emacs   pkg:/editor/gnu-emacs@23.1-0.173.0.0.0.0.487
```

Installing and Updating Packages

The `pkg install` command installs packages that are not currently installed and updates packages that are already installed. The `pkg install` command requires one or more package names.

The `pkg update` command updates installed packages. If you specify a package that is not already installed to the `pkg update` command, the system does not install that package. The `pkg update` command takes zero or more names of packages that are already installed. Specifying no package names updates all packages that are installed in the image. See "Updating All Installed Packages" on page 222.

Installing and updating packages require increased privileges. See "Installation Privileges" in *Adding and Updating Oracle Solaris 11 Software Packages* for more information.

Installing a New Package

Use the commands described in "Getting Information About Packages" on page 216 to identify a package that you want to install. The examples in that section indicated that the `group/feature/amp` package contains Apache, MySQL, and PHP, and that the `group/feature/amp` package is not yet installed in this image. Use the `pkg install` command to install this package.

Tip – Use the `-nv` options to see what the command will do without making any changes to your image.

```
# pkg install -nv group/feature/amp
           Packages to install:         8
     Estimated space available: 112.19 GB
Estimated space to be consumed: 452.42 MB
       Create boot environment:        No
Create backup boot environment:        No
            Services to change:         2
          Rebuild boot archive:        No

Changed packages:
solaris
  database/mysql-51
    None -> 5.1.37,5.11-0.174.0.0.0.0.504:20110920T230125Z
  group/feature/amp
    None -> 0.5.11,5.11-0.174.0.0.0.0.2559:20110921T191255Z
```

```
  web/php-52
    None -> 5.2.17,5.11-0.174.0.0.0.0.504:20110921T041858Z
  web/php-52/extension/php-apc
    None -> 3.0.19,5.11-0.174.0.0.0.0.504:20110921T041245Z
  web/php-52/extension/php-mysql
    None -> 5.2.17,5.11-0.174.0.0.0.0.504:20110921T041411Z
  web/server/apache-22/module/apache-dtrace
    None -> 0.3.1,5.11-0.174.0.0.0.0.504:20110921T042357Z
  web/server/apache-22/module/apache-fcgid
    None -> 2.3.6,5.11-0.174.0.0.0.0.504:20110921T042430Z
  web/server/apache-22/module/apache-php5
    None -> 5.2.17,5.11-0.174.0.0.0.0.504:20110921T042738Z
Services:
  restart_fmri:
    svc:/system/manifest-import:default
    svc:/system/rbac:default
```

The output shows that this command would install eight packages, and the output lists those eight packages in the "Changed packages" section. Comparing this list of "Changed packages" with the list of packages from the `pkg contents` command in "Getting Information About Packages" on page 216 shows that the `web/server/apache-22` package that is part of the group package will not be installed. The following command confirms that this package is already installed and at the correct version.

```
$ pkg list web/server/apache-22
NAME (PUBLISHER)          VERSION                    IFO
web/server/apache-22      2.2.19-0.174.0.0.0.0.504   i--
```

This package could have been installed separately. The following command shows that the package was required by two other packages that are installed.

```
$ pkg search -l -o pkg.name -H ':depend:require:web/server/apache-22'
install/installadm
package/pkg/system-repository
```

The `pkg install -nv` output also shows that installing the `group/feature/amp` package does not create a new boot environment by default. A new boot environment is automatically created when you update particular system packages such as some drivers and other kernel components. A new boot environment might be automatically created when you install, uninstall, or update.

Tip – Explicitly specifying a new boot environment is the safest way to install or update.

The new boot environment is a clone of the current boot environment with the specified install, uninstall, or update changes applied. The current boot environment is not modified. The system is not automatically restarted. The new boot environment is the default boot selection the next time you restart the system. The current boot environment is still available to be booted.

Use the `--be-name` option to force a new boot environment to be created.

```
# pkg install --be-name s11amp group/feature/amp
           Packages to install:   8
       Create boot environment: Yes
Create backup boot environment:  No

DOWNLOAD                                  PKGS       FILES    XFER (MB)
Completed                                  8/8     640/640    70.9/70.9

PHASE                                        ACTIONS
Install Phase                                942/942

PHASE                                          ITEMS
Package State Update Phase                     8/8
Image State Update Phase                       2/2

PHASE                                          ITEMS
Reading Existing Index                         8/8
Indexing Packages                              8/8

A clone of solaris-174 exists and has been updated and activated.
On the next boot the Boot Environment s11amp will be
mounted on '/'.  Reboot when ready to switch to this updated BE.

# pkg list group/feature/amp
pkg list: no packages matching 'group/feature/amp' installed
```

The `pkg list` command reports that the `group/feature/amp` package is not installed because the `group/feature/amp` package is not installed in the current boot environment. The `group/feature/amp` package is installed in the new `s11amp` boot environment.

Use the `beadm list` command to check that the system has a new active boot environment named `s11amp`. The "N" boot environment is currently booted; the "R" boot environment is the default on reboot.

```
# beadm list
BE           Active Mountpoint Space   Policy Created
--           ------ ---------- -----   ------ -------
s11amp       R      -          20.75G  static 2011-09-23 13:58
solaris      -      -          44.81M  static 2010-11-07 17:45
solaris-151a -      -          158.12M static 2010-11-12 14:37
solaris-174  N      /          30.04M  static 2011-09-02 12:38
```

Check that the `group/feature/amp` package is installed in the new boot environment. The "i" in the I column indicates that the `group/feature/amp` package is installed.

```
# beadm mount s11amp /mnt
# pkg -R /mnt list group/feature/amp
NAME (PUBLISHER)       VERSION                    IFO
group/feature/amp      0.5.11-0.174.0.0.0.0.2559    i--
```

Remember to unmount the `s11amp` boot environment.

```
# beadm list
BE           Active Mountpoint Space   Policy Created
--           ------ ---------- -----   ------ -------
```

```
s11amp       R      /mnt       20.75G  static 2011-09-23 13:58
solaris      -      -          44.81M  static 2010-11-07 17:45
solaris-151a -      -          158.12M static 2010-11-12 14:37
solaris-174  N      /          30.05M  static 2011-09-02 12:38
# beadm unmount s11amp
# beadm list
BE           Active Mountpoint Space   Policy Created
--           ------ ---------- -----   ------ -------
s11amp       R      -          20.75G  static 2011-09-23 13:58
solaris      -      -          44.81M  static 2010-11-07 17:45
solaris-151a -      -          158.12M static 2010-11-12 14:37
solaris-174  N      /          30.06M  static 2011-09-02 12:38
```

Updating All Installed Packages

Use one of the following commands to update all currently installed packages that have updates available for this image:

- Do not specify any packages to update.

  ```
  # pkg update --be-name updateBEname
  ```

- Specify `'*'` as the packages to update.

  ```
  # pkg update --be-name updateBEname '*'
  ```

All packages that are installed in the current image and that have updates available for this image are updated.

This operation is likely to update key system packages and create a new boot environment. As a best practice, use the `-nv` option with this command first. Then use the `--be-name` option if necessary to create a new boot environment with the specified name.

◆ ◆ ◆ CHAPTER 13

Managing Disk Use (Tasks)

This chapter describes how to optimize disk space by locating unused files and large directories.

For information on the procedures associated with managing disk use, see "Managing Disk Use (Task Map)" on page 223.

Managing Disk Use (Task Map)

Task	Description	For Instructions
Display information about files and disk space.	Display information about how disk space is used by using the `df` command.	"How to Display Information About Files and Disk Space" on page 225
Display the size of files.	Display information about the size of files by using the `ls` command with the `-lh` options.	"How to Display the Size of Files" on page 227
Find large files.	The `ls -s` command allows you to sort files by size, in descending order.	"How to Find Large Files" on page 228
Find files that exceed a specified size limit.	Locate and display the names of files that exceed a specified size by using the `find` command with the `-size` option and the value of the specified size limit.	"How to Find Files That Exceed a Specified Size Limit" on page 229
Display the size of directories, subdirectories, and files.	Display the size of one or more directories, subdirectories, and files by using the `du` command.	"How to Display the Size of Directories, Subdirectories, and Files" on page 230

Task	Description	For Instructions
List the newest files.	Display the most recently created or changed files first, by using the `ls -t` command.	"How to List the Newest Files" on page 231
Find and remove old or inactive files.	Use the `find` command with the `-atime` and `-mtime` options to locate files that have not been accessed for a specified number of days. You can remove these files by using the `rm` `cat` *filename* command.	"How to Find and Remove Old or Inactive Files" on page 232
Clear out temporary directories.	Locate temp directories, then use the `rm -r *` command to remove the entire directory.	"How to Clear Out Temporary Directories" on page 233
Find and delete core files.	Find and delete core files by using the `find . -name core -exec rm {} \;` command.	"How to Find and Delete `core` Files" on page 234
Delete crash dump files.	Delete crash dump files that are located in the `/var/crash` directory by using the `rm *` command.	"How to Delete Crash Dump Files" on page 234

Displaying Information About Files and Disk Space

This table summarizes the commands available for displaying information about file size and disk space.

Command	Description	Man Page
`df`	Reports the number of free disk blocks and files	`df`(1M)
`du`	Summarizes disk space allocated to each subdirectory	`du`(1)
`find -size`	Searches recursively through a directory based on the size specified with the `-size` option	`find`(1)
`ls -lh`	Lists the size of a file in the power of 1024 scaling	`ls`(1)

▼ How to Display Information About Files and Disk Space

- **Display information about how disk space is used by using the `df` command.**

  `$ df [directory] [-h] [-t]`

`df`	With no options, lists all mounted file systems and their device names, the number of 512-byte blocks used, and the number of files.
directory	Specifies the directory whose file system you want to check.
`-h`	Displays disk space in the power of 1024 scaling.
`-t`	Displays the total blocks as well as the blocks used for all mounted file systems.

Example 13–1 Displaying Information About File Size and Disk Space

In the following example, all the file systems listed are locally mounted except for `/usr/dist`.

```
$ df
/                  (rpool/ROOT/solaris):100709074 blocks 100709074 files
/devices           (/devices          ):       0 blocks        0 files
/dev               (/dev              ):       0 blocks        0 files
/system/contract   (ctfs               ):       0 blocks 2147483601 files
/proc              (proc               ):       0 blocks    29946 files
/etc/mnttab        (mnttab             ):       0 blocks        0 files
/system/volatile   (swap               ):42191440 blocks  2276112 files
/system/object     (objfs              ):       0 blocks 2147483441 files
/etc/dfs/sharetab  (sharefs            ):       0 blocks 2147483646 files
/dev/fd            (fd                 ):       0 blocks        0 files
/tmp               (swap               ):42191440 blocks  2276112 files
/export            (rpool/export       ):100709074 blocks 100709074 files
/export/home       (rpool/export/home  ):100709074 blocks 100709074 files
/export/home/admin (rpool/export/home/admin):100709074 blocks 100709074 files
/rpool             (rpool              ):100709074 blocks 100709074 files
/home/joey    (home.domain:/export/home1/03/joey):960033722 blocks 67158851 files
```

Example 13–2 Displaying File Size Information in 1024 Bytes

In the following example, file system information is displayed in 1024 bytes on one line of information for each specified file system

```
$ df -h
Filesystem             Size   Used  Available Capacity  Mounted on
rpool/ROOT/solaris      67G   2.7G        48G     6%    /
/devices                 0K     0K         0K     0%    /devices
/dev                     0K     0K         0K     0%    /dev
ctfs                     0K     0K         0K     0%    /system/contract
proc                     0K     0K         0K     0%    /proc
mnttab                   0K     0K         0K     0%    /etc/mnttab
```

```
swap                    20G   704K        20G     1%    /system/volatile
objfs                    0K     0K         0K     0%    /system/object
sharefs                  0K     0K         0K     0%    /etc/dfs/sharetab
fd                       0K     0K         0K     0%    /dev/fd
swap                    20G     0K        20G     0%    /tmp
rpool/export            67G    32K        48G     1%    /export
rpool/export/home       67G    32K        48G     1%    /export/home
rpool/export/home/admin
                        67G    33K        48G     1%    /export/home/admin
rpool                   67G    74K        48G     1%    /rpool
home.domain:/export/home1/03/joey
                       539G    81G       452G    16%    /home/joey
```

Example 13–3 Displaying Total Number of Blocks and Files Allocated for a File System

The following example shows a list of all mounted file systems, device names, total 512-byte blocks used, and the number of files. The second line of each two-line entry displays the total number of blocks and files that are allocated for the file system.

```
$ df -t
/                  (rpool/ROOT/solaris): 100709077 blocks 100709077 files
                                 total: 140378112 blocks 100838460 files
/devices           (/devices           ):        0 blocks        0 files
                                 total:        0 blocks      456 files
/dev               (/dev               ):        0 blocks        0 files
                                 total:        0 blocks      681 files
/system/contract   (ctfs               ):        0 blocks 2147483601 files
                                 total:        0 blocks       46 files
/proc              (proc               ):        0 blocks    29946 files
                                 total:        0 blocks    30002 files
/etc/mnttab        (mnttab             ):        0 blocks        0 files
                                 total:        0 blocks        1 files
/system/volatile   (swap               ): 42190928 blocks  2276112 files
                                 total: 42192336 blocks  2276330 files
/system/object     (objfs              ):        0 blocks 2147483441 files
                                 total:        0 blocks      206 files
/etc/dfs/sharetab  (sharefs            ):        0 blocks 2147483646 files
                                 total:        0 blocks        1 files
/dev/fd            (fd                 ):        0 blocks        0 files
                                 total:        0 blocks       31 files
/tmp               (swap               ): 42190928 blocks  2276112 files
                                 total: 42190928 blocks  2276330 files
/export            (rpool/export       ): 100709077 blocks 100709077 files
                                 total: 140378112 blocks 100709085 files
/export/home       (rpool/export/home  ): 100709077 blocks 100709077 files
                                 total: 140378112 blocks 100709085 files
/export/home/admin (rpool/export/home/admin): 100709077 blocks 100709077 files
                                 total: 140378112 blocks 100709086 files
/rpool             (rpool              ): 100709077 blocks 100709077 files
                                 total: 140378112 blocks 100709090 files
/home/joey    (home.domain:/export/home1/03/joey): 960033724 blocks 67158850 files
                                 total: 1129776786 blocks 67966080 files
```

Checking the Size of Files

You can check the size of files and sort them by using the `ls` command. You can find files that exceed a size limit by using the `find` command. For more information, see the `ls(1)` and `find(1)` man pages.

Note – If you run out of space in the `/var` directory, do not symbolically link the `/var` directory to a directory on a file system with more disk space. Doing so, even as a temporary measure, might cause problems for certain daemon processes and utilities.

▼ How to Display the Size of Files

1 Change to the directory where the files you want to check are located.

2 Display the size of the files.

```
$ ls [-lh] [-s]
```

`-l` Displays a list of files and directories in long format, showing the sizes in bytes. (See the example that follows.)

`-h` Scales file sizes and directory sizes into Kbytes, Mbytes, Gbytes, or Tbytes when the file or directory size is larger than 1024 bytes. This option also modifies the output displayed by the `-o`, `-n`, `-@`, and `-g` options to display file or directory sizes in the new format. For more information, see the `ls(1)` man page.

`-s` Displays a list of the files and directories, showing the sizes in blocks.

Example 13–4 Displaying the Size of Files

The following example shows that the `lastlog` and `messages` files are larger than the other files in the `/var/adm` directory.

```
$ cd /var/adm
$ ls -lh
total 682
drwxrwxr-x   9 root     sys           15 Jan  4 03:10 .
drwxr-xr-x  41 root     sys           42 Jan  3 19:03 ..
drwxrwxr-x   5 adm      adm            5 Jan  3 17:26 acct
-rw-------   1 uucp     bin            0 Jan  3 17:34 aculog
drwxr-xr-x   2 adm      adm            2 Jan  3 17:26 exacct
-r--r--r--   1 root     root        3.2M Jan  4 06:47 lastlog
drwxr-xr-x   2 adm      adm            2 Jan  3 17:26 log
-rw-r--r--   1 root     root           0 Jan  4 03:10 messages
-rw-r--r--   1 root     root         55K Jan  3 19:10 messages.0
drwxr-xr-x   2 root     sys            2 Jan  3 17:26 pool
drwxrwxr-x   2 adm      sys            2 Jan  3 17:26 sa
```

```
drwxr-xr-x   2 root     sys            2 Jan  3 17:26 sm.bin
drwxr-xr-x   2 root     sys            2 Jan  3 17:26 streams
lrwxrwxrwx   1 root     root          27 Jan  3 17:36 utmpx -> \
 ../../system/volatile/utmpx
-rw-r--r--   1 adm      adm          10K Jan  4 06:47 wtmpx
```

The following example shows that the `lpsched.1` file uses two blocks.

```
$ cd /var/lp/logs
$ ls -s
total 2            0 lpsched       2 lpsched.1
```

How to Find Large Files

1. **Change to the directory that you want to search.**

2. **Display the size of files in blocks from largest to smallest.**

   - If the characters or columns for the files are *different*, use the following command to sort a list of files by block size, from largest to smallest.

     ```
     $ ls -l | sort +4rn | more
     ```

     Note that this command sorts files in a list by the character that is in the fourth field, starting from the left.

   - If the characters or columns for the files are the *same*, use the following command to sort a list of files by block size, from largest to smallest.

     ```
     $ ls -s | sort -nr | more
     ```

     Note that this command sorts files in a list, starting with the left most character.

Example 13–5 Finding Large Files (Sorting by the Fifth Field's Character)

```
$ cd /var/adm
$ ls -l | sort +4rn | more
-r--r--r--   1 root     root     3353420 Jan  7 06:45 lastlog
-rw-r--r--   1 root     root     3221924 Jan  7 12:15 messages
-rw-r--r--   1 root     root       56045 Jan  3 19:10 messages.0
-rw-r--r--   1 adm      adm        12648 Jan  7 06:45 wtmpx
drwxr-xr-x  41 root     sys           42 Jan  3 19:03 ..
lrwxrwxrwx   1 root     root          27 Jan  3 17:36 utmpx -> ../../system/volatile/utmpx
drwxrwxr-x   9 root     sys           15 Jan  4 03:10 .
drwxrwxr-x   5 adm      adm            5 Jan  3 17:26 acct
drwxr-xr-x   2 adm      adm            2 Jan  3 17:26 exacct
drwxr-xr-x   2 adm      adm            2 Jan  3 17:26 log
drwxr-xr-x   2 root     sys            2 Jan  3 17:26 pool
drwxr-xr-x   2 root     sys            2 Jan  3 17:26 sm.bin
drwxr-xr-x   2 root     sys            2 Jan  3 17:26 streams
drwxrwxr-x   2 adm      sys            2 Jan  3 17:26 sa
```

```
-rw-------   1 uucp     bin            0 Jan  3 17:34 aculog
```

Example 13–6 Finding Large Files (Sorting by the Left Most Character)

In the following example, the `lastlog` and `messages` files are the largest files in the `/var/adm` directory.

```
$ cd /var/adm
$ ls -s | sort -nr | more
 6409 -rw-r--r--   1 root     root     3221924 Jan  7 12:15 messages
 517 -r--r--r--   1 root     root     3353420 Jan  7 06:45 lastlog
 111 -rw-r--r--   1 root     root       56045 Jan  3 19:10 messages.0
  26 -rw-r--r--   1 adm      adm        12648 Jan  7 06:45 wtmpx
   5 drwxr-xr-x  41 root     sys           42 Jan  3 19:03 ..
   3 drwxrwxr-x   9 root     sys           15 Jan  4 03:10 .
   3 drwxrwxr-x   5 adm      adm            5 Jan  3 17:26 acct
   3 drwxrwxr-x   2 adm      sys            2 Jan  3 17:26 sa
   3 drwxr-xr-x   2 root     sys            2 Jan  3 17:26 streams
   3 drwxr-xr-x   2 root     sys            2 Jan  3 17:26 sm.bin
   3 drwxr-xr-x   2 root     sys            2 Jan  3 17:26 pool
   3 drwxr-xr-x   2 adm      adm            2 Jan  3 17:26 log
   3 drwxr-xr-x   2 adm      adm            2 Jan  3 17:26 exacct
   1 lrwxrwxrwx   1 root     root          27 Jan  3 17:36 utmpx -> ../../system/volatile/utmpx
   1 -rw-------   1 uucp     bin            0 Jan  3 17:34 aculog
total 7094
```

▼ How to Find Files That Exceed a Specified Size Limit

- **To locate and display the names of files that exceed a specified size, use the `find` command.**

  `$ find` *directory* `-size +`*nnn*

  directory Identifies the directory that you want to search.

  `-size +`*nnn* Is a number of 512-byte blocks. Files that exceed this size are listed.

Example 13–7 Finding Files That Exceed a Specified Size Limit

The following example shows how to find files larger than 400 blocks in the current working directory. The `-print` option displays the output of the `find` command.

```
$ find . -size +400 -print
./Howto/howto.doc
./Howto/howto.doc.backup
./Howto/howtotest.doc
./Routine/routineBackupconcepts.doc
./Routine/routineIntro.doc
./Routine/routineTroublefsck.doc
./.record
./Mail/pagination
./Config/configPrintadmin.doc
```

```
./Config/configPrintsetup.doc
./Config/configMailappx.doc
./Config/configMailconcepts.doc
./snapshot.rs
```

Checking the Size of Directories

You can display the size of directories by using the du command and options. For more information about these commands, see the du(1) man page.

▼ How to Display the Size of Directories, Subdirectories, and Files

- **Display the size of one or more directories, subdirectories, and files by using the du command. Sizes are displayed in 512-byte blocks.**

  ```
  $ du [-as] [directory ...]
  ```

du	Displays the size of each directory that you specify, including each subdirectory beneath it.
-a	Displays the size of each file and subdirectory, and the total number of blocks that are contained in the specified directory.
-s	Displays the total number of blocks that are contained in the specified directory.
-h	Displays the size of each directory in 1024–byte blocks.
-H	Displays the size of each directory in 1000–byte blocks.
[*directory ...*]	Identifies one or more directories that you want to check. Separate multiple directories in the command-line syntax with spaces.

Example 13–8 Displaying the Size of Directories, Subdirectories, and Files

The following example shows the sizes of two directories.

```
$ du -s /var/adm /var/spool/cups
7098    /var/adm
0       /var/spool/cups
```

The following example shows the sizes of two directories and includes the sizes of all the subdirectories and files that are contained within each directory. The total number of blocks that are contained in each directory is also displayed.

```
$ du /var/adm /var/spool/cups
3       /var/adm/streams
3       /var/adm/sa
3       /var/adm/acct/fiscal
3       /var/adm/acct/nite
3       /var/adm/acct/sum
12      /var/adm/acct
3       /var/adm/exacct
3       /var/adm/sm.bin
3       /var/adm/log
3       /var/adm/pool
7098    /var/adm
```

The following example shows directory sizes in 1024–byte blocks.

```
$ du -h /usr/share/audio
 796K   /usr/share/audio/samples/au
 797K   /usr/share/audio/samples
 798K   /usr/share/audio
```

Finding and Removing Old or Inactive Files

Part of the job of cleaning up heavily loaded file systems involves locating and removing files that have not been used recently. You can locate unused files by using the `ls` or `find` commands. For more information, see the `ls(1)` and `find(1)` man pages.

Other ways to conserve disk space include emptying temporary directories such as the directories located in `/var/tmp` or `/var/spool`, and deleting `core` and crash dump files. For more information about crash dump files, refer to Chapter 17, Managing System Crash Information (Tasks).

▼ How to List the Newest Files

- **List files, displaying the most recently created or changed files first, by using the `ls -t` command.**

  `$ ls -t` [*directory*]

  `-t` Sorts files by latest time stamp first.

  directory Identifies the directory that you want to search.

Example 13–9 Listing the Newest Files

The following example shows how to use the `ls -tl` command to locate the most recently created or changed files within the `/var/adm` directory. The `sulog` file was created or edited most recently.

```
$ ls -tl /var/adm
-rw-r--r--   1 root     root     3227516 Jan  7 12:22 messages
-rw-r--r--   1 adm      adm        12648 Jan  7 06:45 wtmpx
-r--r--r--   1 root     root     3353420 Jan  7 06:45 lastlog
drwxrwxr-x   9 root     sys           15 Jan  4 03:10 .
-rw-r--r--   1 root     root       56045 Jan  3 19:10 messages.0
drwxr-xr-x  41 root     sys           42 Jan  3 19:03 ..
lrwxrwxrwx   1 root     root          27 Jan  3 17:36 utmpx -> ../../system/volatile/utmpx
-rw-------   1 uucp     bin            0 Jan  3 17:34 aculog
drwxr-xr-x   2 root     sys            2 Jan  3 17:26 streams
drwxr-xr-x   2 root     sys            2 Jan  3 17:26 sm.bin
drwxrwxr-x   2 adm      sys            2 Jan  3 17:26 sa
drwxr-xr-x   2 root     sys            2 Jan  3 17:26 pool
drwxr-xr-x   2 adm      adm            2 Jan  3 17:26 log
drwxr-xr-x   2 adm      adm            2 Jan  3 17:26 exacct
drwxrwxr-x   5 adm      adm            5 Jan  3 17:26 acct
```

▼ How to Find and Remove Old or Inactive Files

1 Become an administrator.

For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2 Find files that have not been accessed for a specified number of days and list them in a file.

```
# find directory -type f[-atime +nnn] [-mtime +nnn] -print > filename &
```

directory	Identifies the directory you want to search. Directories below this directory are also searched.
`-atime +`*nnn*	Finds files that have not been accessed within the number of days (*nnn*) that you specify.
`-mtime +`*nnn*	Finds files that have not been modified within the number of days (*nnn*) that you specify.
filename	Identifies the file that contains the list of inactive files.

3 Remove the inactive files found listed in the previous step.

```
# rm `cat filename`
```

where *filename* identifies the file that was created in the previous step. This file contains the list of inactive files.

Example 13–10 Finding and Removing Old or Inactive Files

The following example shows files in the `/var/adm` directory and the subdirectories that have not been accessed in the last 60 days. The `/var/tmp/deadfiles` file contains the list of inactive files. The `rm` command removes these inactive files.

```
# find /var/adm -type f -atime +60 -print > /var/tmp/deadfiles &
# more /var/tmp/deadfiles
/var/adm/aculog
/var/adm/spellhist
/var/adm/wtmpx
/var/adm/sa/sa13
/var/adm/sa/sa27
/var/adm/sa/sa11
/var/adm/sa/sa23
/var/adm/sulog
/var/adm/vold.log
/var/adm/messages.1
/var/adm/messages.2
/var/adm/messages.3
# rm ‘cat /var/tmp/deadfiles‘
#
```

▼ How to Clear Out Temporary Directories

1. **Become an administrator.**

   For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*.

2. **Change to the directory that you want to clean out.**

   `# cd` *directory*

Caution – Ensure that you are in the correct directory before completing Step 3. Step 3 deletes all files in the current directory.

3. **Delete the files and subdirectories in the current directory.**

   ```
   # rm -r *
   ```

4. **Change to other directories that contain unnecessary, temporary or obsolete subdirectories and files.**

5. **Delete these subdirectories and files by repeating Step 3.**

Example 13–11 Clearing Out Temporary Directories

The following example shows how to clear out the `mywork` directory, and how to verify that all files and subdirectories were removed.

```
# cd mywork
# ls
filea.000
```

```
fileb.000
filec.001
# rm -r *
# ls
#
```

How to Find and Delete `core` Files

1. **Become an administrator.**
2. **Change to the directory where you want to search for `core` files.**
3. **Find and remove any `core` files in this directory and its subdirectories.**

```
# find . -name core -exec rm {} \;
```

Example 13–12 Finding and Deleting `core` Files

The following example shows how to find and remove `core` files from the `jones` user account by using the `find` command.

```
# cd /home/jones
# find . -name core -exec rm {} \;
```

How to Delete Crash Dump Files

Crash dump files can be very large. If you have enabled your system to store these files, do not retain them for longer than necessary.

1. **Become an administrator.**
2. **Change to the directory where crash dump files are stored.**

```
# cd /var/crash/
```

Caution – Ensure you are in the correct directory before completing Step 3. Step 3 deletes all files in the current directory.

3. **Remove the crash dump files.**

```
# rm *
```

4. **Verify that the crash dump files were removed.**

```
# ls
```

Example 13–13 Deleting Crash Dump Files

The following example shows how to remove crash dump files from the system `venus`, and how to verify that the crash dump files were removed.

```
# cd /var/crash
# rm *
# ls
```

◆ ◆ ◆ CHAPTER 14

Scheduling System Tasks (Tasks)

This chapter describes how to schedule routine or single (one-time) system tasks by using the `crontab` and `at` commands.

This chapter also explains how to control access to these commands by using the following files:

- `cron.deny`
- `cron-allow`
- `at.deny`

This is a list of the information that is in this chapter:

- "Creating and Editing `crontab` Files (Task Map)" on page 237
- "Using the `at` Command (Task Map)" on page 249

Creating and Editing `crontab` Files (Task Map)

Task	Description	For Instructions
Create or edit a `crontab` file.	Use the `crontab -e` command to create or edit a `crontab` file.	"How to Create or Edit a `crontab` File" on page 243
Verify that a `crontab` file exists.	Use the `ls -l` command to verify the contents of the `/var/spool/cron/crontabs` file.	"How to Verify That a `crontab` File Exists" on page 244
Display a `crontab`file.	Use the `ls -l` command to display the `crontab` file.	"How to Display a `crontab` File" on page 244
Remove a `crontab` file.	The `crontab` file is set up with restrictive permissions Use the `crontab -r` command, rather than the `rm` command to remove a `crontab` file.	"How to Remove a `crontab` File" on page 245

Task	Description	For Instructions
Deny `crontab` access.	To deny users access to `crontab` commands, add user names to the `/etc/cron.d/cron.deny` file by editing this file.	"How to Deny `crontab` Command Access" on page 247
Limit `crontab` access to specified users.	To allow users access to the `crontab` command, add user names to the `/etc/cron.d/cron.allow` file.	"How to Limit `crontab` Command Access to Specified Users" on page 248

Ways to Automatically Execute System Tasks

You can set up many system tasks to execute automatically. Some of these tasks should occur at regular intervals. Other tasks need to run only once, perhaps during off hours such as evenings or weekends.

This section contains overview information about two commands, `crontab` and `at`, which enable you to schedule routine tasks to execute automatically. The `crontab` command schedules repetitive commands. The `at` command schedules tasks that execute once.

The following table summarizes `crontab` and `at` commands, as well as the files that enable you to control access to these commands.

TABLE 14–1 Command Summary: Scheduling System Tasks

Command	What It Schedules	Location of Files	Files That Control Access
`crontab`	Multiple system tasks at regular intervals	`/var/spool/cron/crontabs`	`/etc/cron.d/cron.allow` and `/etc/cron.d/cron.deny`
`at`	A single system task	`/var/spool/cron/atjobs`	`/etc/cron.d/at.deny`

For Scheduling Repetitive Jobs: `crontab`

You can schedule routine system administration tasks to execute daily, weekly, or monthly by using the `crontab` command.

Daily `crontab` system administration tasks might include the following:

- Removing files more than a few days old from temporary directories
- Executing accounting summary commands
- Taking snapshots of the system by using the `df` and `ps` commands
- Performing daily security monitoring

- Running system backups

Weekly `crontab` system administration tasks might include the following:

- Rebuilding the `catman` database for use by the `man -k` command
- Running the `fsck -n` command to list any disk problems

Monthly `crontab` system administration tasks might include the following:

- Listing files not used during a specific month
- Producing monthly accounting reports

Additionally, users can schedule `crontab` commands to execute other routine system tasks, such as sending reminders and removing backup files.

For step-by-step instructions on scheduling `crontab` jobs, see "How to Create or Edit a `crontab` File" on page 243.

For Scheduling a Single Job: `at`

The `at` command allows you to schedule a job for execution at a later time. The job can consist of a single command or a script.

Similar to `crontab`, the `at` command allows you to schedule the automatic execution of routine tasks. However, unlike `crontab` files, `at` files execute their tasks once. Then, they are removed from their directory. Therefore, the `at` command is most useful for running simple commands or scripts that direct output into separate files for later examination.

Submitting an `at` job involves typing a command and following the `at` command syntax to specify options to schedule the time your job will be executed. For more information about submitting `at` jobs, see "Description of the `at` Command" on page 250.

The `at` command stores the command or script you ran, along with a copy of your current environment variable, in the `/var/spool/cron/atjobs` directory. Your `at` job file name is given a long number that specifies its location in the `at` queue, followed by the `.a` extension, such as `793962000.a`.

The `cron` daemon checks for `at` jobs at startup and listens for new jobs that are submitted. After the `cron` daemon executes an `at` job, the `at` job's file is removed from the `atjobs` directory. For more information, see the `at(1)` man page.

For step-by-step instructions on scheduling `at` jobs, see "How to Create an `at` Job" on page 251.

Scheduling a Repetitive System Task (cron)

The following sections describe how to create, edit, display, and remove crontab files, as well as how to control access to them.

Inside a crontab File

The cron daemon schedules system tasks according to commands found within each crontab file. A crontab file consists of commands, one command per line, that will be executed at regular intervals. The beginning of each line contains date and time information that tells the cron daemon when to execute the command.

For example, a crontab file named root is supplied during SunOS software installation. The file's contents include these command lines:

```
10 3 * * * /usr/sbin/logadm      (1)
15 3 * * 0 /usr/lib/fs/nfs/nfsfind      (2)
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1      (3)
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean      (4)
```

The following describes the output for each of these command lines:

- The first line runs the logadm command at 3:10 a.m. every day.
- The second line executes the nfsfind script every Sunday at 3:15 a.m.
- The third line runs a script that checks for daylight savings time (and make corrections, if necessary) at 2:10 a.m. daily.

  If there is no RTC time zone, nor an /etc/rtc_config file, this entry does nothing.

  x86 only – The /usr/sbin/rtc script can only be run on an x86 based system.

- The fourth line checks for (and removes) duplicate entries in the Generic Security Service table, /etc/gss/gsscred_db, at 3:30 a.m. daily.

For more information about the syntax of lines within a crontab file, see "Syntax of crontab File Entries" on page 241.

The crontab files are stored in the /var/spool/cron/crontabs directory. Several crontab files besides root are provided during SunOS software installation. See the following table.

TABLE 14–2 Default crontab Files

crontab File	Function
adm	Accounting

TABLE 14–2 Default crontab Files *(Continued)*

crontab File	Function
root	General system functions and file system cleanup
sys	Performance data collection
uucp	General uucp cleanup

Besides the default crontab files, users can create crontab files to schedule their own system tasks. Other crontab files are named after the user accounts in which they are created, such as bob, mary, smith, or jones.

To access crontab files that belong to root or other users, superuser privileges are required.

Procedures explaining how to create, edit, display, and remove crontab files are described in subsequent sections.

How the cron Daemon Handles Scheduling

The cron daemon manages the automatic scheduling of crontab commands. The role of the cron daemon is to check the /var/spool/cron/crontab directory for the presence of crontab files.

The cron daemon performs the following tasks at startup:

- Checks for new crontab files.
- Reads the execution times that are listed within the files.
- Submits the commands for execution at the proper times.
- Listens for notifications from the crontab commands regarding updated crontab files.

In much the same way, the cron daemon controls the scheduling of at files. These files are stored in the /var/spool/cron/atjobs directory. The cron daemon also listens for notifications from the crontab commands regarding submitted at jobs.

Syntax of crontab File Entries

A crontab file consists of commands, one command per line, that execute automatically at the time specified by the first five fields of each command line. These five fields, described in the following table, are separated by spaces.

TABLE 14–3 Acceptable Values for crontab Time Fields

Time Field	Values
Minute	0-59
Hour	0-23
Day of month	1-31
Month	1-12
Day of week	0-6 (0 = Sunday)

Follow these guidelines for using special characters in crontab time fields:

- Use a space to separate each field.
- Use a comma to separate multiple values.
- Use a hyphen to designate a range of values.
- Use an asterisk as a wildcard to include all possible values.
- Use a comment mark (#) at the beginning of a line to indicate a comment or a blank line.

For example, the following crontab command entry displays a reminder in the user's console window at 4 p.m. on the first and fifteenth days of every month.

```
0 16 1,15 * * echo Timesheets Due > /dev/console
```

Each command within a crontab file must consist of one line, even if that line is very long. The crontab file does not recognize extra carriage returns. For more detailed information about crontab entries and command options, refer to the crontab(1) man page.

Creating and Editing crontab Files

The simplest way to create a crontab file is to use the crontab -e command. This command invokes the text editor that has been set for your system environment. The default editor for your system environment is defined in the EDITOR environment variable. If this variable has not been set, the crontab command uses the default editor, ed. Preferably, you should choose an editor that you know well.

The following example shows how to determine if an editor has been defined, and how to set up vi as the default.

```
$ which $EDITOR
$
$ EDITOR=vi
$ export EDITOR
```

When you create a crontab file, it is automatically placed in the /var/spool/cron/crontabs directory and is given your user name. You can create or edit a crontab file for another user, or root, if you have superuser privileges.

▼ How to Create or Edit a crontab File

Before You Begin If you are creating or editing a crontab file that belongs to root or another user you must become root.

You do not need to become root to edit your own crontab file.

1 Create a new crontab file, or edit an existing file.

```
# crontab -e [username]
```

where *username* specifies the name of the user's account for which you want to create or edit a crontab file. You can create your own crontab file without superuser privileges, but you must have superuser privileges to creating or edit a crontab file for root or another user.

Caution – If you accidentally type the crontab command with no option, press the interrupt character for your editor. This character allows you to quit without saving changes. If you instead saved changes and exited the file, the existing crontab file would be overwritten with an empty file.

2 Add command lines to the crontab file.

Follow the syntax described in "Syntax of crontab File Entries" on page 241. The crontab file will be placed in the /var/spool/cron/crontabs directory.

3 Verify your crontab file changes.

```
# crontab -l [username]
```

Example 14–1 Creating a crontab File

The following example shows how to create a crontab file for another user.

```
# crontab -e jones
```

The following command entry added to a new crontab file automatically removes any log files from the user's home directory at 1:00 a.m. every Sunday morning. Because the command entry does not redirect output, redirect characters are added to the command line after *.log. Doing so ensures that the command executes properly.

```
# This command helps clean up user accounts.
1 0 * * 0 rm /home/jones/*.log > /dev/null 2>&1
```

▼ How to Verify That a crontab File Exists

- **To verify that a crontab file exists for a user, use the ls -l command in the /var/spool/cron/crontabs directory. For example, the following output shows that crontab files exist for users jones and smith.**

  ```
  $ ls -l /var/spool/cron/crontabs
  ```

  Verify the contents of user's crontab file by using the crontab -l command as described in "How to Display a crontab File" on page 244.

Displaying crontab Files

The crontab -l command displays the contents of a crontab file much the same way that the cat command displays the contents of other types of files. You do not have to change the directory to /var/spool/cron/crontabs directory (where crontab files are located) to use this command.

By default, the crontab -l command displays your own crontab file. To display crontab files that belong to other users, you must be superuser.

▼ How to Display a crontab File

Before You Begin Become the root user to display a crontab file that belongs to root or another user.

You do not need to become root to display your own crontab file.

- **Display the crontab file.**

  ```
  # crontab -l [username]
  ```

  where *username* specifies the name of the user's account for which you want to display a crontab file. Displaying another user's crontab file requires superuser privileges.

Caution – If you accidentally type the crontab command with no option, press the interrupt character for your editor. This character allows you to quit without saving changes. If you instead saved changes and exited the file, the existing crontab file would be overwritten with an empty file.

Example 14–2 Displaying a crontab File

This example shows how to use the crontab -l command to display the contents of the user's default crontab file.

```
$ crontab -l
13 13 * * * chmod g+w /home1/documents/*.book > /dev/null 2>&1
```

Example 14–3 Displaying the Default `root` `crontab` file.

This example shows how to display the default `root` `crontab` file.

```
$ suPassword:

# crontab -l
#ident  "@(#)root       1.19    98/07/06 SMI"   /* SVr4.0 1.1.3.1       */
#
# The root crontab should be used to perform accounting data collection.
#
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
#10 3 * * * /usr/lib/krb5/kprop_script ___slave_kdcs___
```

Example 14–4 Displaying the `crontab` File of Another User

This example shows how to display the `crontab` file that belongs to another user.

```
$ su
Password:
# crontab -l jones
13 13 * * * cp /home/jones/work_files /usr/backup/. > /dev/null 2>&1
```

Removing `crontab` Files

By default, `crontab` file protections are set up so that you cannot inadvertently delete a `crontab` file by using the `rm` command. Instead, use the `crontab -r` command to remove `crontab` files.

By default, the `crontab -r` command removes your own `crontab` file.

You do not have to change the directory to `/var/spool/cron/crontabs` (where `crontab` files are located) to use this command.

▼ How to Remove a `crontab` File

Before You Begin Become the `root` user to remove a `crontab` file that belongs to `root` or another user. Roles contain authorizations and privileged commands.

You do not need to become `root` to remove your own `crontab` file.

1 Remove the `crontab` file.

```
# crontab -r [username]
```

where *username* specifies the name of the user's account for which you want to remove a `crontab` file. Removing `crontab` files for another user requires superuser privileges.

Caution – If you accidentally type the `crontab` command with no option, press the interrupt character for your editor. This character allows you to quit without saving changes. If you instead saved changes and exited the file, the existing `crontab` file would be overwritten with an empty file.

2 Verify that the `crontab` file has been removed.

```
# ls /var/spool/cron/crontabs
```

Example 14–5 Removing a `crontab` File

The following example shows how user `smith` uses the `crontab -r` command to remove his own `crontab` file.

```
$ ls /var/spool/cron/crontabs
adm     jones     root    smith    sys     uucp
$ crontab -r
$ ls /var/spool/cron/crontabs
adm     jones    root    sys    uucp
```

Controlling Access to the `crontab` Command

You can control access to the `crontab` command by using two files in the `/etc/cron.d` directory: `cron.deny` and `cron.allow`. These files permit only specified users to perform `crontab` command tasks such as creating, editing, displaying, or removing their own `crontab` files.

The `cron.deny` and `cron.allow` files consist of a list of user names, one user name per line.

These access control files work together as follows:

- If `cron.allow` exists, only the users who are listed in this file can create, edit, display, or remove `crontab` files.
- If `cron.allow` does not exist, all users can submit `crontab` files, except for users who are listed in `cron.deny`.
- If neither `cron.allow` nor `cron.deny` exists, superuser privileges are required to run the `crontab` command.

Superuser privileges are required to edit or create the `cron.deny` and `cron.allow` files.

The `cron.deny` file, which is created during SunOS software installation, contains the following user names:

```
$ cat /etc/cron.d/cron.deny
daemon
bin
```

```
smtp
nuucp
listen
nobody
noaccess
```

None of the user names in the default `cron.deny` file can access the `crontab` command. You can edit this file to add other user names that will be denied access to the `crontab` command.

No default `cron.allow` file is supplied. So, after Oracle Solaris software installation, all users (except users who are listed in the default `cron.deny` file) can access the `crontab` command. If you create a `cron.allow` file, only these users can access the `crontab` command.

▼ How to Deny `crontab` Command Access

1 Become the root role.

```
$ su -
Password:
#
```

Note – This method works whether `root` is a user or a role.

---

2 Edit the `/etc/cron.d/cron.deny` file and add user names, one user per line. Include users who will be denied access to the `crontab` commands.

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
username1
username2
username3
.
.
.
```

3 Verify that the `/etc/cron.d/cron.deny` file contains the new entries.

```
# cat /etc/cron.d/cron.deny
daemon
bin
nuucp
listen
nobody
noaccess
```

▼ How to Limit crontab Command Access to Specified Users

1. **Become the root role.**
2. **Create the `/etc/cron.d/cron.allow` file.**
3. **Add the `root` user name to the `cron.allow` file.**

   If you do not add `root` to the file, superuser access to `crontab` commands will be denied.
4. **Add the user names, one user name per line.**

   Include users that will be allowed to use the `crontab` command.

   ```
   root
   username1
   username2
   username3
   .
   .
   .
   ```

Example 14–6 Limiting `crontab` Command Access to Specified Users

The following example shows a `cron.deny` file that prevents user names `jones`, `temp`, and `visitor` from accessing the `crontab` command.

```
$ cat /etc/cron.d/cron.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
jones
temp
visitor
```

The following example shows a `cron.allow` file. The users `root`, `jones`, and `smith` are the only users who can access the `crontab` command.

```
$ cat /etc/cron.d/cron.allow
root
jones
smith
```

How to Verify Limited crontab Command Access

To verify if a specific user can access the crontab command, use the crontab -l command while you are logged into the user account.

```
$ crontab -l
```

If the user can access the crontab command, and already has created a crontab file, the file is displayed. Otherwise, if the user can access the crontab command but no crontab file exists, a message similar to the following message is displayed:

```
crontab: can't open your crontab file
```

Either this user either is listed in the cron.allow file (if the file exists), or the user is not listed in the cron.deny file.

If the user cannot access the crontab command, the following message is displayed whether or not a previous crontab file exists:

```
crontab: you are not authorized to use cron. Sorry.
```

This message means that either the user is not listed in the cron.allow file (if the file exists), or the user is listed in the cron.deny file.

Using the at Command (Task Map)

Task	Description	For Instructions
Create an at job.	Use the atcommand to do the following: ■ Start the at utility from the command line. ■ Type the commands or scripts that you want to execute, one per line. ■ Exit the at utility and save the job.	"How to Create an at Job" on page 251
Display the at queue.	User the atq command to display the at queue.	"How to Display the at Queue" on page 252
Verify an at job.	Use the atq command to confirm that at jobs that belong to a specific user have been submitted to the queue.	"How to Verify an at Job" on page 252

Task	Description	For Instructions
Display at jobs.	Use the at -l [*job-id*] command to display at jobs that have been submitted to the queue.	"How to Display at Jobs" on page 252
Remove at jobs.	Use the at -r [*job-id*] command to remove at jobs from the queue.	"How to Remove at Jobs" on page 253
Deny access to the at command.	To deny users access to the atcommand, edit the /etc/cron.d/at.deny file.	"How to Deny Access to the at Command" on page 254

Scheduling a Single System Task (at)

The following sections describe how to use the at command to perform the following tasks:

- Schedule jobs (command and scripts) for execution at a later time
- How to display and remove these jobs
- How to control access to the at command

By default, users can create, display, and remove their own at job files. To access at files that belong to root or other users, you must have superuser privileges.

When you submit an at job, it is assigned a job identification number along with the .a extension. This designation becomes the job's file name, as well as its queue number.

Description of the at Command

Submitting an at job file involves these steps:

1. Invoking the at utility and specifying a command execution time.
2. Typing a command or script to execute later.

Note – If output from this command or script is important, be sure to direct the output to a file for later examination.

For example, the following at job removes core files from the user account smith near midnight on the last day of July.

```
$ at 11:45pm July 31
at> rm /home/smith/*core*
at> Press Control-d
commands will be executed using /bin/csh
job 933486300.a at Tue Jul 31 23:45:00 2004
```

Controlling Access to the at Command

You can set up a file to control access to the at command, permitting only specified users to create, remove, or display queue information about their at jobs. The file that controls access to the at command, /etc/cron.d/at.deny, consists of a list of user names, one user name per line. The users who are listed in this file cannot access at commands.

The at.deny file, which is created during SunOS software installation, contains the following user names:

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
```

With superuser privileges, you can edit the at.deny file to add other user names whose at command access you want to restrict.

▼ How to Create an at Job

1 Start the at utility, specifying the time you want your job executed.

```
$ at [-m] time [date]
```

`-m` Sends you email after the job is completed.

time Specifies the hour that you want to schedule the job. Add am or pm if you do not specify the hours according to the 24-hour clock. Acceptable keywords are midnight, noon, and now. Minutes are optional.

date Specifies the first three or more letters of a month, a day of the week, or the keywords today or tomorrow.

2 At the at prompt, type the commands or scripts that you want to execute, one per line.

You may type more than one command by pressing Return at the end of each line.

3 Exit the at utility and save the at job by pressing Control-D.

Your at job is assigned a queue number, which is also the job's file name. This number is displayed when you exit the at utility.

Example 14–7 Creating an at Job

The following example shows the at job that user jones created to remove her backup files at 7:30 p.m. She used the -m option so that she would receive an email message after her job completed.

```
$ at -m 1930
at> rm /home/jones/*.backup
at> Press Control-D
job 897355800.a at Thu Jul  12 19:30:00 2004
```

She received a email message which confirmed the execution of her at job.

```
Your "at" job "rm /home/jones/*.backup"
completed.
```

The following example shows how jones scheduled a large at job for 4:00 a.m. Saturday morning. The job output was directed to a file named big.file.

```
$ at 4 am Saturday
at> sort -r /usr/dict/words > /export/home/jones/big.file
```

How to Display the at Queue

- **To check your jobs that are waiting in the at queue, use the atq command.**

  ```
  $ atq
  ```

  This command displays status information about the at jobs that you have created.

How to Verify an at Job

- **To verify that you have created an at job, use the atq command. In the following example, the atq command confirms that at jobs that belong to jones have been submitted to the queue.**

  ```
  $ atq
  Rank      Execution Date     Owner     Job         Queue   Job Name
    1st   Jul 12, 2004 19:30   jones  897355800.a     a     stdin
    2nd   Jul 14, 2004 23:45   jones  897543900.a     a     stdin
    3rd   Jul 17, 2004 04:00   jones  897732000.a     a     stdin
  ```

How to Display at Jobs

- **To display information about the execution times of your at jobs, use the at -l command.**

  ```
  $ at -l [job-id]
  ```

where the -l *job-id* option identifies the identification number of the job whose status you want to display.

Example 14–8 Displaying at Jobs

The following example shows output from the at -l command, which provides information about the status of all jobs submitted by a user.

```
$ at -l
897543900.a    Sat Jul 14 23:45:00 2004
897355800.a    Thu Jul 12 19:30:00 2004
897732000.a    Tue Jul 17 04:00:00 2004
```

The following example shows the output that is displayed when a single job is specified with the at -l command.

```
$ at -l 897732000.a
897732000.a        Tue Jul 17 04:00:00 2004
```

▼ How to Remove at Jobs

Before You Begin Become the root user to remove an at job that belongs to root or another user. Roles contain authorizations and privileged commands.

You do not need to become root to remove you own at job.

1 Remove the at job from the queue before the job is executed.

```
# at -r [job-id]
```

where the -r *job-id* option specifies the identification number of the job you want to remove.

2 Verify that the at job is removed by using the at -l (or the atq) command.

The at -l command displays the jobs remaining in the at queue. The job whose identification number you specified should not appear.

```
$ at -l [job-id]
```

Example 14–9 Removing at Jobs

In the following example, a user wants to remove an at job that was scheduled to execute at 4 a.m. on July 17th. First, the user displays the at queue to locate the job identification number. Next, the user removes this job from the at queue. Finally, the user verifies that this job has been removed from the queue.

```
$ at -l
897543900.a    Sat Jul 14 23:45:00 2003
```

```
897355800.a    Thu Jul 12 19:30:00 2003
897732000.a    Tue Jul 17 04:00:00 2003
$ at -r 897732000.a
$ at -l 897732000.a
at: 858142000.a: No such file or directory
```

▼ How to Deny Access to the at Command

1. **Become the root role.**
2. **Edit the /etc/cron.d/at.deny file and add the names of users, one user name per line, that will be prevented from using the at commands.**

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
username1
username2
username3
.
.
.
```

Example 14–10 Denying at Access

The following example shows an at.deny file that has been edited so that the users smith and jones cannot access the at command.

```
$ cat at.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
jones
smith
```

How to Verify That at Command Access Is Denied

- **To verify that a username was added correctly to the /etc/cron.d/at.deny file, use the at -l command while logged in as the user. If the user smith cannot access the at command, the following message is displayed:**

```
# su smith
Password:
# at -l
at: you are not authorized to use at.  Sorry.
```

Likewise, if the user tries to submit an at job, the following message is displayed:

```
# at 2:30pm
at: you are not authorized to use at.  Sorry.
```

This message confirms that the user is listed in the at.deny file.

If at command access is allowed, then the at -l command returns nothing.

◆ ◆ ◆ CHAPTER 15

Setting Up and Administering Printers by Using CUPS (Tasks)

This chapter includes information for managing your printing environment by using the Common UNIX Printing System (CUPS), including how to transition to using CUPS if you previously used the LP print service to manage printers. The interfaces that are described in this chapter include the CUPS command-line utilities, the CUPS web browser interface, and CUPS Print Manager, a GUI that can be accessed in the desktop.

This is a list of the information that is in this chapter:

- "Introduction to CUPS" on page 257
- "Setting Up Your Printing Environment to Work With CUPS" on page 260
- "Setting Up and Administering Printers by Using CUPS Command-Line Utilities" on page 263
- "Setting Up and Administering Printers by Using the CUPS Web Browser Interface (Task Map)" on page 271
- "Setting Up Printers by Using CUPS Print Manager (Task Map)" on page 277
- "Administering Printers by Using CUPS Print Manager (Task Map)" on page 283

Introduction to CUPS

CUPS is a modular and open-source printing system that uses the Internet Printing Protocol (IPP) as the basis for managing printers, print requests, and print queues. CUPS supports network printer browsing and PostScript Printer Description-based printing options. CUPS also provides a common printing interface across a local network.

IPP is the standard protocol for printing on a network. Similar to other IP-based protocols, IPP can be used locally or over the Internet to communicate with remote printers. Unlike other protocols, IPP also supports access control, authentication, and encryption, making it a much more capable and secure printing solution than other protocols. IPP is layered on top of the Hypertext Transfer Protocol (HTTP). HTTP is the basis for web servers that are on the Internet. When IPP is in use, you can verify printer or server status information and manage printers and

print jobs through a browser. CUPS is a complete IPP/1.1 based printing system that provides basic, digest, and local certificate authentication and user, domain, or IP-based access control.

CUPS includes support for dynamic printer detection and grouping. CUPS replaces the `lpr` command with its own command and the LPD printer drivers with its own print drivers. CUPS is similar to the LP print service in that it uses PostScript format as its underlying language for page descriptions. Because CUPS provides both the System V and Berkeley print commands, users and applications can print to CUPS queues with little or no changes to the options that were previously used.

Lastly, CUPS includes application-level interfaces that are used by many open-source applications and toolkits. On the back end, CUPS includes the necessary interfaces for processing the annotated raster image format (RIP). Support for this format and these interfaces is integrated into other critical open-source print driver technologies.

CUPS is the default and the only print service in the Oracle Solaris release, replacing the LP print service. Printing in the Oracle Solaris operating system (OS) by using CUPS is managed by using the following:

- CUPS command-line utilities – These commands include new CUPS print commands, as well as some print commands that were previously used by the LP print service.
- CUPS web browser interface – Go to `http://localhost:631`.
- CUPS Print Manager GUI – You can access the GUI from the Oracle Solaris Desktop, which includes GNOME 2.30, or by typing the `system-config-printer` command in a terminal window.

CUPS Processes

For CUPS to manage your printing environment, you must first create a print queue under CUPS. The print queue might point to a printer that is connected directly to your system through a USB port or a parallel port. However, the queue can also point to a printer on the network, a printer on the Internet, or multiple printers, depending on how you have configured the application. Regardless of where the queue points, the print queue is treated like any other printer.

CUPS Services

CUPS services are provided through two new Service Management Facility (SMF) services:

- `svc:/application/cups/scheduler`

  This service manages the `cupsd` daemon. This daemon provides basic printing services that include queueing, filtering, spooling, notification, IPP support, device enumeration, and web management.

- `svc:/application/cups/in-lpd`

  This service runs the `cupsd-lpd` daemon. This daemon provides basic RFC-1179 (LPD protocol) support for the CUPS service.

The Printer Management profile and the `solaris.smf.manage.cups` authorization enable users who do not have a `root` login to manage these SMF services.

Setting Up Printers and Print Queues by Using CUPS

For CUPS to manage your printing environment, you must first create a print queue under CUPS.

You can create a new print queue in one of the following ways:

- Use the `lpadmin` command to manually create the print queue. For more information, see the `lpadmin`(8) man page.
- Use the Print Manager GUI, which can be accessed from the Oracle Solaris Desktop. For more information, see "Setting Up Printers by Using CUPS Print Manager (Task Map)" on page 277.
- Use the web browser interface. After you install CUPS, go to `http://localhost:631/admin`.
- Physically connect a USB printer to your local system.

  If CUPS is enabled on your system, the hardware abstraction layer (HAL) and the `hal-cups-utils` utility recognize the USB printer hot-plug events. They can recognize new printers that are connected to your local system. The `hal-cups-utils` utility automatically creates a print queue under CUPS for the new printer.

  In addition, CUPS supports printer discovery by using the mDNS framework (Bonjour) and SNMP. CUPS can discover printers that are shared by other CUPS servers through the CUPS browsing feature. For more information, go to `http://www.cups.org/documentation.php/doc-1.5/options.html`.
- For network print queues, enable the CUPS "browse feature" (the default) on your system. If another system on the network advertises an available printer on the remote system, CUPS detects the printer, and a new print queue is created.

Managing Print Requests by Using CUPS

Every time you submit a print request, CUPS creates a print job that contains information about the print queue to which you are sending the request, the name of the document, and the page description. Print jobs are numbered, for example, `queue-1`, `queue-2`, so that you can monitor each print job as it is printed or cancel the print job, if necessary.

When a print request is submitted, CUPS does the following:

1. Determines which programs to use (filters, print drivers, port monitors, and back-end programs).
2. Runs these programs to complete the print job.
3. Removes the job from the print queue when the print job is complete, and then prints the next print job that is submitted. You can configure CUPS to notify you when a print job is complete or if any errors occur during printing.

Setting Up Your Printing Environment to Work With CUPS

In previous Oracle Solaris releases, the LP print service was the default print service. Starting with the Oracle Solaris 11 release, the LP print service is removed. The default and only available print service in Oracle Solaris 11 is CUPS. If you are performing a fresh installation of Oracle Solaris 11 and have any existing printers that were configured by using the LP print service, you need to reconfigure those printers by using CUPS after the installation.

If you are upgrading from Oracle Solaris 11 Express to Oracle Solaris 11, see "How to Set Up Your Printing Environment" on page 261.

Switching to the CUPS print environment has resulted in the following changes:

- Any existing printers that were configured by using the LP print service will no longer work and must be reconfigured.

  You can reconfigure printers by using any one of the following methods:

  - By using the `lpadmin` command. For information, see "How to Set Up a Printer by Using the `lpadmin` Command" on page 264.
  - By using the CUPS web browser interface at `http://localhost:631/help`. For information, see "Setting Up and Administering Printers by Using the CUPS Web Browser Interface " on page 272.
  - By using CUPS Print Manager. For information, see "Setting Up Printers by Using CUPS Print Manager" on page 278.
- Printer configuration that was previously stored in the NIS naming service is not used by CUPS. Administrators can share network printers that are configured by using the CUPS shared-printer feature. CUPS auto-discovers printers on a network and enables you to print to these printers without any manual configuration. For information about sharing printers by using CUPS Print Manager, see "Remote Server Configuration" on page 278
- Printers that are configured on a per-user basis in the `~/.printers` file no longer work. Printer configuration is managed by using the CUPS web browser interface, the CUPS command-line utilities, or the CUPS Print Manager graphical user interface.
- In previous releases, the `/etc/printers.conf` file contained details about all of the printers that were added by using the LP print service. With the removal of the LP print service in the Oracle Solaris 11 OS, this file still exists under CUPS but contains a summary of the local print queues. After installing the OS, any information about printers that were previously

configured by using `lp` print commands is removed. The resulting behavior is as though these printers were never configured on the system. Any existing printers must be reconfigured by using CUPS. You do not need to delete existing printers prior to reconfiguring these printers by using CUPS. For information about setting up your printing environment to work with CUPS, see "How to Set Up Your Printing Environment" on page 261.

How to Set Up Your Printing Environment

To transition your current printing environment to work with CUPS, you must reconfigure your existing printers.

1. **Ensure that the `cups/scheduler` and the `cups/in-lpd` SMF services are online.**

```
$ svcs -a | grep cups/scheduler
online 18:18:55 svc:/application/cups/scheduler:default

$ svcs -a | grep cups/in-lpd
online Sep_29 svc:/application/cups/in-lpd:default
```

2. **To enable these services, type the following commands:**

```
# svcadm enable cups/scheduler
# svcadm enable cups/in-lpd
```

3. **Determine whether the `printer/cups/system-config-printer` package is installed on your system.**

```
$ pkg info print/cups/system-config-printer
```

   - **If the package is already installed, configure your printer by using CUPS.**

     Printers can be configured by using either the `lpadmin` command, or by using the CUPS web browser interface at `http://localhost:631`, or by using CUPS Print Manager, which is accessible in the desktop.

   - **If the package is not installed, install the package.**

```
$ pkg install print/cups/system-config-printer
```

See Also Additional CUPS documentation can be found at:

- `http://www.cups.org/documentation.php`
- `http://www.cups.org/doc-1.1/sam.html`

Next Steps You can now configure printers by using CUPS. You can set a default printer by specifying either the `LPDEST` or `PRINTER` environment variables, or by using the `lpoptions` command. For instructions, see "How to Set a Default Printer at the Command Line" on page 266 and Example 15–6.

Setting Up Your Printing Environment for an Upgrade

If you are running unmodified Oracle Solaris 11 Express, CUPS is already the default print service. If you upgrade to Oracle Solaris 11, you do not need to reconfigure any existing print queues by using CUPS. However, if you switched to the LP print service and configured printers by using the `lp` print commands, you must reconfigure these existing printers by using CUPS after the upgrade.

Caution – If you are running the LP print service, ensure to back up your `/etc/printers.conf` file before upgrading, as the upgrade process removes this file.

To determine which print service is enabled on your system, type the following command:

```
$ /usr/sbin/print-service -q
```

Reconfigure your existing printers by using any one of the methods that is described in this chapter.

Setting Up and Administering Printers by Using CUPS Command-Line Utilities (Task Map)

Task	Description	For Instructions
Set up a new locally attached printer.	You can set up a new locally attached printer by using the `lpadmin` command.	"How to Set Up a Printer by Using the `lpadmin` Command" on page 264
Set the default printer for a system at the command line.	You can set a default printer destination for a user by specifying the `LPDEST` and `PRINTER` environment variables, and by using the `lpoptions` command.	"How to Set a Default Printer at the Command Line" on page 266
Verify the status of printers.	You can verify the status of all printers or a specific printer by using the `lpstat` command. This command enables you to determine which printers are available for use and enables you to examine the characteristics of those printers.	"How to Verify the Status of Printers" on page 268
Print a file by using CUPS commands.	You can print a file by using the `lp` and `lpr` commands.	"How to Print a File to the Default Printer" on page 269

Task	Description	For Instructions
Delete a printer and remove printer access.	You can delete a printer and remove printer access by using the `lpoptions` command.	"How to Delete a Printer and Remove Printer Access" on page 270

Setting Up and Administering Printers by Using CUPS Command-Line Utilities

This section provides a brief description of the CUPS commands and describes how to set up and administer your printers.

CUPS Command-Line Utilities

CUPS provides various commands to set up printers and make those printers accessible to systems on the network. In addition, CUPS supports several printer-specific options that enable you to control printer configuration. The following table lists frequently used CUPS commands.

Note – Some CUPS command names are the same as legacy LP print commands, but the behavior of commands under CUPS management might be different.

TABLE 15–1 CUPS Command-Line Utilities

Command	Task
`cancel`(1)	Cancels a print request
`cuspaccept`(8)	Enables queueing of print requests to the named destinations
`cuspdisable`(8)	Disables the named printers or classes
`cupsenable`(8)	Enables the named printers or classes
`cupsreject`(8)	Rejects queueing of print requests to the named destinations
`lp`(1)	Submits a print request
`lpadmin`(8)	Sets up or changes a printer or class configuration
`lpc`(8)	Provides limited control over CUPS print and class queues
`lpinfo`(8)	Shows available devices or drivers known to the CUPS server
`lpmove`(8)	Moves a specified job or all jobs to a new destination
`lpoptions`(1)	Displays or sets printer options and defaults

TABLE 15–1 CUPS Command-Line Utilities *(Continued)*

Command	Task
`lpq`(1)	Shows the current print queue status
`lpr`(1)	Submits a print request
`lprm`(1)	Cancels print jobs that have been queued for printing
`lpstat`(1)	Displays the status information for queues and requests

▼ How to Set Up a Printer by Using the `lpadmin` Command

1 Connect the printer to the system, then turn on the power to the printer.

Consult the printer vendor's installation documentation for information about hardware switches and cabling requirements.

2 Become an administrator.

For more information, see "How to Obtain Administrative Rights" in *Oracle Solaris Administration: Security Services*

3 Use the `lpadmin` command with the `-p` option to add a printer to CUPS.

Only the most commonly used options of the CUPS `lpadmin` command are shown here. For information about other options, see the `lpadmin`(8) man page.

```
$ /usr/sbin/lpadmin -p printer-name -E -v device -m ppd
```

`-p` Specifies the name of the printer to add.

`-E` Enables the destination and accepts jobs.

`-v` Sets the `device-uri` attribute of the print queue.

`-m` Sets the PPD file for the printer from the model directory or by using one of the driver interfaces.

See the examples at the end of this procedure.

4 Enable the printer to accept print requests and to print those requests.

```
$ cupsaccept printer-name
$ cupsenable printer-name
```

5 Verify that the printer is correctly configured.

```
$ lpstat -p printer-name -l
```

Example 15–1 Adding a Printer That Is Connected to the Parallel Port

To add an HP DeskJet printer `DeskJet` that is connected to the parallel port, you would type the following command:

```
$ /usr/sbin/lpadmin -p DeskJet -E -v parallel:/dev/lp1 -m deskjet.ppd
```

`deskjet.ppd` PPD file for the HP DeskJet drivers included with CUPS

Example 15–2 Adding a Printer That Uses a PPD File

To add an HP LaserJet printer `LaserJet` by using a JetDirect network interface with the IP address `10.1.1.1`, you would type the following command:

```
$ /usr/sbin/lpadmin -p LaserJet -E -v socket://10.1.1.1 -m laserjet.ppd
```

`laserjet.ppd` PPD file for the HP LaserJet drivers included with CUPS

Example 15–3 Adding a Printer That Is Connected to the Serial Port

To add a dot matrix printer that is connected to the serial port, you would type the following command:

```
$/usr/sbin/lpadmin -p DotMatrix -E -m epson9.ppd \
-v serial:/dev/ttyS0?baud=9600+size=8+parity=none+flow=soft
```

Specify the serial port, baud rate, number of bits, parity, and flow control. If you do not need flow control, delete the `+flow=soft` attribute.

Setting a Default Printer

You can specify the default printer in one of the following ways:

- By setting the `LPDEST` or `PRINTER` environment variable.

  The `LPDEST` environment variable determines the destination of the printer. If the `LPDEST` variable is not set, the `PRINTER` variable is used. The `PRINTER` variable determines the output device or destination. If both the `LPDEST` and `PRINTER` variables are not set, an unspecified device is used. For instructions on setting up a default printer by specifying the environment variables, see "How to Set a Default Printer at the Command Line" on page 266.

- By using the new `lpoptions` command.

  Use this command to display or set printer options and defaults. For instructions on setting up a default printer by using the CUPS commands, see "How to Set a Default Printer at the Command Line" on page 266. For more information, see the `lpoptions`(1) man page.

The print command searches for the default printer in the following order:

1. The printer name as set by the `lp` command with the `-d` option
2. The value of the `LPDEST` environment variable
3. The value of the `PRINTER` environment variable

For instructions on setting up printers by using the CUPS web browser interface, see "Setting Up and Administering Printers by Using the CUPS Web Browser Interface " on page 272.

How to Set a Default Printer at the Command Line

The default printer can be a local printer or a remote printer.

1 Become an administrator on the system where you want to set a default printer.

2 Set the system's default printer by using one of the following methods:

- **By specifying the `PRINTER` variable:**

  `$ export PRINTER=`*printer-name*

  where *printer-name* specifies the name of the printer to be assigned as the system's default printer. If you do not specify *printer-name*, the system is set up with no default printer.

  Note – When using the `lp` command with the `-d` option, the destination printer, which might not be the default printer, is specified. If the `-d` option is not specified, the `print` command searches for information about the printer in the `PRINTER` environment variable.

- **By specifying the `LPDEST` variable:**

  `$ export LPDEST=`*printer-name*

  where *printer-name* specifies the name of the printer to be assigned as the system's default printer. If you do not specify *printer-name*, the system is set up with no default printer.

  Note – If both the `LPDEST` and the `PRINTER` environment variables are set, `LPDEST` takes precedence.

- **By using the `lpoptions` command:**

  `$ lpoptions -d` *printer-name*

`-d`	Specifies the destination printer.
printer-name	Specifies the name of the printer that is assigned as the system's default printer. If you do not specify *printer-name*, the system is set up with no default printer.

For more information, see the `lpoptions`(1) man page.

3 **Verify the system's default printer.**

```
$ lpstat -d
```

4 **To print to the default printer, type the following command:**

```
$ lp filename
```

Example 15–4 Setting a Default Printer by Specifying the `PRINTER` Variable

The following example shows how to set the printer `luna` as the system's default printer by using the `PRINTER` variable.

```
$ export PRINTER=luna
$ lpstat -d
system default destination: luna
```

Example 15–5 Setting a Default Printer by Specifying the `LPDEST` Variable

The following example shows how to set the printer `luna` as the system's default printer by specifying the `LPDEST` variable.

```
$ export LPDEST=luna
$ lpstat -d
system default destination: luna
```

Example 15–6 Setting a Default Printer by Using the `lpoptions` Command

The following example shows how to set the printer `luna` as the system's default printer. The printer `luna` is used as the system's default printer if the `LPDEST` or the `PRINTER` environment variable is not set.

```
$ lpoptions -d luna
$ lpstat -d
system default destination: luna
```

The `lpoptions` command creates a `~/.lpoptions` file that includes and entry for the default printer `luna` in the file. By default, all print jobs are now directed to the `luna` printer.

▼ How to Print to a Specified Printer

1 **(Optional) Verify the status of the printer.**

```
$ lpstat -p printer-name
```

2 Provide the destination printer name when issuing the `lp` command.

`$ lp -d` *destination-printer filename*

`-d`	Specifies the destination printer.
destination-printer	Specifies the name of the printer that you are assigning as the destination printer.
filename	Specifies the file name to print.

Note – You can also use the `lpr` command with the `-p` option to submit a print request to a specific printer. For more information, see the `lpr`(1) man page.

Example 15–7 Printing to a Specified Printer by Using the `lp` Command

The following example shows how to set the printer `luna` as the destination printer.

```
$ lp -d luna abc.ps
request id is luna-1 (1 file(s))

$ lpstat -d
system default destination: saturn
```

The `-d` option of the `lp` command takes precedence over the `LPDEST` and `PRINTER` environment variables.

Note that in this example, the default printer is `saturn`.

▼ How to Verify the Status of Printers

The `lpstat` command displays information about accessible printers and jobs.

1 Log in to any system on the network.

2 (Optional) Verify the status of all printers or a specific printer.

Only the most commonly used options are shown here. For information about other options, see the `lpstat`(1) man page.

`$ lpstat [-d] [-p]` *printer-name* `[-l] [-t]`

`-d`	Shows the system's default printer.
`-p` *printer-name*	Shows that a printer is active or idle, and when the printer was enabled or disabled.

You can specify multiple printer names with this command. Use a space or a comma to separate printer names. If you use spaces, enclose the list of printer names in quotation marks. If you do not specify *printer-name*, the status of all printers is displayed.

`-l` Shows the characteristics of printers and jobs.

`-t` Shows status information about CUPS, including the status of all printers, for example whether printers are active and accepting print requests.

Example 15–8 Displaying the Status of Printers

To display the status of the printer `luna`:

```
$ lpstat -p luna
printer luna is idle. enabled since Jul 12 11:17 2011. available.
```

To display the system's default printer:

```
$ lpstat -d
system default destination: luna
```

To display the description of the printers `asteroid` and `luna`:

```
$ lpstat -p "asteroid, luna" -D
printer asteroid faulted. enabled since Jan 5 11:35 2011. available.
unable to print: paper misfeed jam

Description: Printer by break room
printer luna is idle. enabled since Jan 5 11:36 2011. available.
Description: Printer by server room.
```

To display the characteristics of the printer `luna`:

```
$ lpstat -p luna -l
printer luna is idle.  enabled since September 29, 2011 05:20:57 PM BST
```

▼ How to Print a File to the Default Printer

1. **Log in to any system on the network.**
2. **(Optional) Verify the status of the printer.**

   $ **lpstat -p** *printer-name*

3 **Issue a print request in one of the following ways:**

- **By using the `lp` command:**

  `$ lp` *filename*

- **By using the `lpr` command:**

  `$ lpr` *filename*

Note – Only the basic commands are shown in this procedure. For information about the other options, see the `lp`(1) and the `lpr`(1) man pages.

How to Delete a Printer and Remove Printer Access

1 **Become an administrator on a print client with access to the printer to delete.**

2 **On the system that is the print client, delete information about the printer.**

`$ lpoptions -x` *printer-name*

printer-name	Specifies the name of the printer to delete.
`-x`	Deletes the specified printer.

Note – The `-x` option only removes the default options for a specific printer and instance. The original print queue still remains until it is deleted by using the `lpadmin` command.

---

3 **Become an administrator.**

4 **On the system that is the printer server, stop accepting print requests for the printer.**

`$ cupsreject` *printer-name*

This step prevents any new requests from entering the printer's queue while you are in the process of removing the printer.

5 **Stop the printer.**

`$ cupsdisable` *printer-name*

6 **Delete the printer.**

`$ lpadmin -x` *printer-name*

7 **Verify that the printer has been deleted, as follows:**

a. **Confirm that the printer has been deleted on the print client.**

```
$ lpstat -p printer-name -l
```

The command output displays a message indicating the printer does not exist.

b. **Confirm that the printer has been deleted on the print server.**

```
$ lpstat -p printer-name -l
```

The command output displays a message indicating that the printer does not exist.

Example 15–9 Deleting a Printer

The following example shows how to delete the printer `luna` from the print client `terra` and from the print server `jupiter`.

```
terra# lpoptions -x luna
terra# lpstat -p luna -l
jupiter# lpadmin -x luna
jupiter# lpstat -p luna -l
lpstat: Invalid destination name in list "luna"!
```

Setting Up and Administering Printers by Using the CUPS Web Browser Interface (Task Map)

Task	Description	For Instructions
Verify requirements for using the web browser interface.	To access the CUPS web browser interface, the CUPS service must be enabled on your system, and CUPS packages must also be installed on your system.	"Requirements for Using the CUPS Web Browser Interface" on page 272
Add a new printer by using the CUPS web browser interface.	Use the Administration tab of the CUPS Print Manager GUI when you connect a new printer to your local system.	"How to Add a New Printer" on page 276

Setting Up and Administering Printers by Using the CUPS Web Browser Interface

You can use the CUPS web browser GUI to manage your printing environment in Oracle Solaris 11. This section describes the requirements to use the web browser interface and the administration tasks that you can perform.

Requirements for Using the CUPS Web Browser Interface

To access the web browser interface, go to `http://localhost:631`. The CUPS web browser interface can be accessed from all supported browsers. Depending on the task that you are performing, you might be prompted for a user name and password, or for the `root` user name and password.

Note the following requirements for using the CUPS web browser interface:

- The CUPS software packages must be installed on the host that is accessing the CUPS web pages. If you are running the Oracle Solaris 11 release, these software packages are installed on your system by default.

  The following CUPS packages are required:

  - `cups`
  - `cups-libs`
  - `foomatic-db`
  - `foomatic-db-engine`
- The CUPS scheduler, `svc:/application/cups/scheduler`, must also be running on the host.

  To verify that the CUPS scheduler is running, open a terminal window and type the following command:

  ```
  $ svcs cups/scheduler
  STATE          STIME    FMRI
  online         10:07:54 svc:/application/cups/scheduler:default
  ```
- The JavaScript scripting language must be supported and enabled on the browser that you are using to access the CUPS web pages.

  Most current browsers support the use of the JavaScript language. To determine whether the JavaScript language is enabled, verify the Content tab of your browser's Preferences menu.

Troubleshooting Issues With Accessing the CUPS Web Browser Interface

If you encounter an error while attempting to access the CUPS web browser interface or you cannot access the interface, see "Requirements for Using the CUPS Web Browser Interface" on page 272 to ensure that all of the requirements have been met. In addition, verify your browser's proxy settings to determine whether a proxy server has been configured. If so, try disabling the proxy server, then re-attempt to access the CUPS web browser interface.

To determine whether the CUPS web browser interface is running, you can also attempt to connect to the CUPS port (Port 631) by typing the `telnet` command in a terminal window, as follows:

```
mymachine% telnet localhost 631
Trying ::1...
Connected to mymachine
Escape character is ^].
^]q
telnet> q
Connection to mymachine closed.
mymachine%
```

To stop the `telnet` session, press the `Control-]`. To quit the `telnet` session, type `q`.

Print Administration Tasks

Common print administration tasks that you can perform by using the CUPS web browser interface include the following:

- Customizing a print server setup
- Pointing a print client to a common print server
- Setting up and managing directly-attached printers and printer classes on servers
- Setting up and managing remote printers and printer classes on servers
- Managing print jobs from print clients

When you first access the CUPS web browser interface at `http://localhost:631`, you see the Home tab. From this tab, you can access all of the print administration tasks, which are grouped together by category, as well as the full set of CUPS documentation.

The following tabs are displayed on the web browser interface's main web page:

- Administration – Enables you to access most print administration tasks, including CUPS server configuration.

  You can directly access the Administration section of the web browser interface by going to `http://localhost:631/admin`.

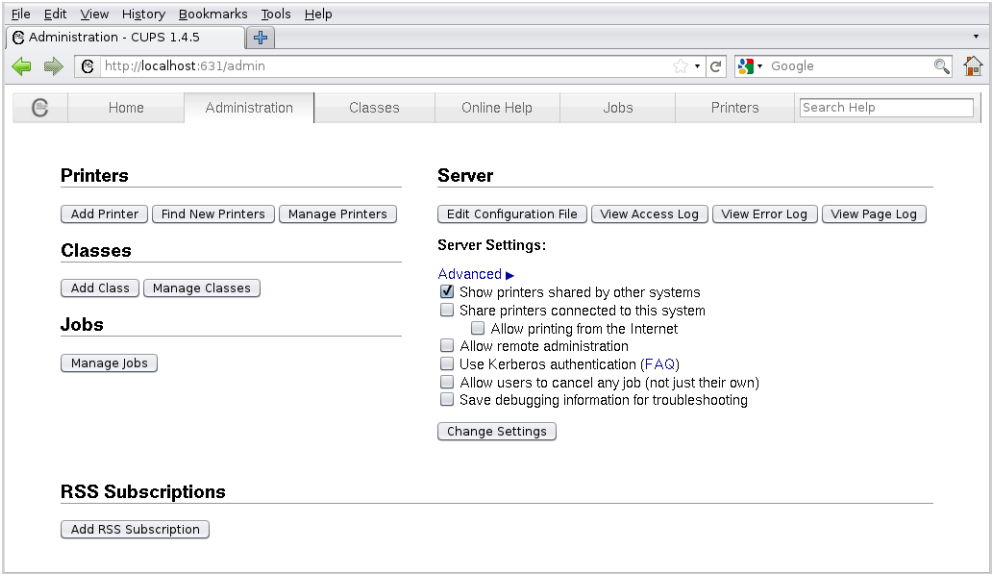
- Classes – Enables you to search printer classes.

  CUPS provides collections of printers, which are called *printer classes*. Print jobs that are sent to a class are forwarded to the first available printer in that class. Classes can be members of other classes. Therefore, you can define very large, distributed printer classes for high-availability printing.
- Documentation – Enables you to access the CUPS documentation, which includes manuals, system administration documentation, FAQs, and online help.
- Jobs – Enables you to view and manage print jobs for configured printers.
- Printers – Enables you to view information about and modify the settings of a specified printer.

About the Administration Tab

Most printing tasks can be performed from the Administration tab. Note that some tasks can be performed from multiple tabs. Basic server settings can also be changed from the Administration tab. For more information about CUPS server configuration, see the `cupsd.conf`(5) man page.

The following figure shows the contents of the Administration tab of the CUPS web browser interface.



The following table describes the task categories and individual tasks that can be performed from the Administration tab.

Task Category	Task Type
Printers	■ Add Printer ■ Find New Printers ■ Manage Printers
Classes	■ Add Class ■ Manage Classes
Jobs	■ Manage Jobs
Server	■ Edit Configuration File ■ View Page Log

About the Printers Tab

The Printers tab enables you to view and modify information for configured print queues, as illustrated in the following figure.

From the Printers tab, you can also perform the following tasks:

- Print a test page
- Stop the printer
- Reject a print job
- Move a print job
- Cancel all print jobs
- Unpublish the printer
- Modify a printer

- Set printer options
- Delete a printer
- Set the printer as the default
- Set allowed users for a printer

▼ How to Add a New Printer

1. **Access the Administration tab by going to `http://localhost:631/admin`.**
2. **Click the Add Printer button.**
3. **If prompted, type your login user name and password, or the `root` user name and password.**
4. **Follow the prompts to complete the process.**

About the CUPS Print Manager GUI

CUPS support includes a GUI, `system-config-printer`, which is accessible from the command line or from the desktop. Because CUPS is the default print service, detection of directly-attached printers is automatic. CUPS can also automatically discover other CUPS printers on a network, if those printers have sharing enabled. CUPS can also be configured to browse the network for Windows-hosted printers. For more information, see "Local Server Configuration" on page 278.

Note that when using CUPS Print Manager to perform a privileged action, such as creating a new print queue, modifying print queue properties, or deleting an existing print queue, you are prompted for the `root` password.

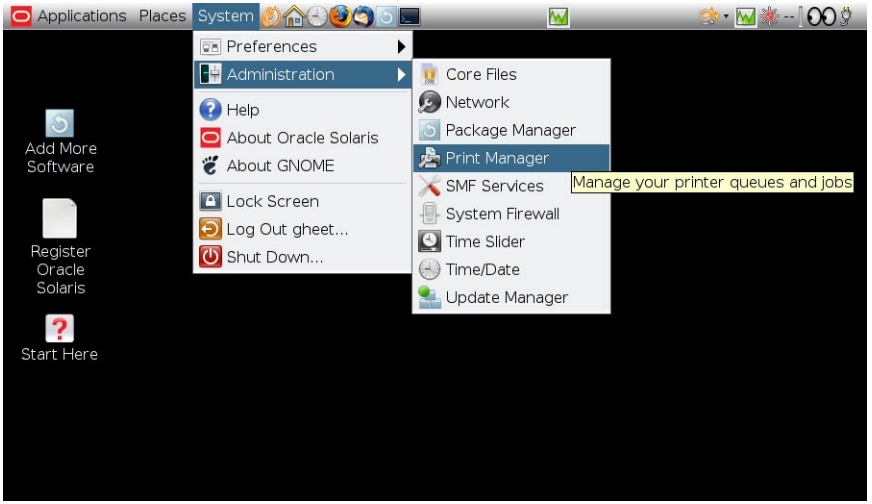
Starting CUPS Print Manager

To start the CUPS Print Manager GUI, use any one of the following methods:

- From the command line, type the following command:

  ```
  $ system-config-printer
  ```

- From the desktop's Main Menubar, choose System → Administration → Print Manager.



Setting Up Printers by Using CUPS Print Manager (Task Map)

Task	Description	For Information
Configure a CUPS server to administer local print queues.	You can use CUPS Print Manager to configure a local server. The system acts as a local server for any printers that are physically connected to it and for any print queues that are created on that system.	"Local Server Configuration" on page 278
Configure a CUPS server to administer remote print queues.	You can use CUPS Print Manager to connect to a remote system, where you can administer print queues. The remote system must be configured to allow remote administration.	"Remote Server Configuration" on page 278
Set up a new local printer.	When you connect a new printer to your local system, the printer is automatically detected and its Properties dialog opens. From here, you can finish configuring the new printer by using the CUPS Print Manager GUI .	"How to Set Up a New Local Printer" on page 281

Setting Up Printers by Using CUPS Print Manager

This section describes tasks that are required to set up printers by using CUPS Print Manager.

Local Server Configuration

Each system that uses CUPS can be both a print server and a print client. The system acts as the server for any printer that is physically connected to it and for any print queues that are created on that system.

To configure advanced settings and options for a local CUPS server, start CUPS Print Manager, then choose Server → Settings. In the Basic Server Settings dialog, click the Advanced button. The following advanced server settings and options can be viewed or configured:

- Job History – Controls the print job history for a specified printer or printers
- Browse Servers – Enables you to restrict browsing by CUPS to certain print servers for the purpose of polling print queues

In the Basic Server Settings dialog you can configure the following settings:

- Show printers that are shared by other systems – Enables other CUPS print queues to be visible to the local system.
- Publish shared printers connected to this system – Publishes a list of print queues that are configured on a system to the other systems on a local area network (LAN). You can also publish print queues to be accessible beyond the LAN. The Allow Printing from the Internet option is only available if this setting has been selected.
- Allow remote administration – Enables you to administer the print queue from a remote system by using either CUPS Print Manager or the CUPS web browser interface.
- Allow users to cancel any job (not just their own) – Enables users to cancel any print jobs.
- Save debugging information for troubleshooting – Enables the logging of debugging information for troubleshooting purposes.

Remote Server Configuration

You can configure CUPS to administer print queues on a remote print server. Typically, you can connect to remote servers within the same local area network (LAN). Only those print queues that are owned by the remote print server to which you are connected can be modified. Each remote server determines whether its configured print queues can be shared or remotely modified based on the Settings dialog for the specified printer.

Published printers are printers that are publicly announced by the server on the LAN, based on how the `cupsd.conf` file has been configured for browsing. Shared or published printers can be detected by remote print clients but unshared or unpublished printers are not announced on the network.

Note – You must have appropriate authorizations to administer remote print queues. In the Oracle Solaris 11 release, you must provide the `root` password for the remote server.

How to Configure CUPS to Administer Remote Print Queues

1. **Start the CUPS Print Manager GUI by choosing System → Administration → Print Manager from the desktop's main menubar or by typing the following command in a terminal window:**

   ```
   $ system-config-printer
   ```

2. **From the Server menu, choose Settings.**

   The Basic Server Settings dialog is displayed.

3. **Select the following options:**

   - Publish Shared Printers Connected to This System – Displays the shared or published printers that can be detected by remote print clients. If you do not enable this option, some printers might not be displayed in the list of available printers when you connect to the remote server.
   - Allow Remote Administration – Enables you to connect to a remote server.

4. **Click OK.**

5. **From the Server menu, choose the Connect option.**

   The Connect to CUPS Server dialog is displayed.

6. **Select the desired remote server from the CUPS server list.**

7. **(Optional) If you require encryption, select the Require Encryption option.**

8. **Click the Connect button.**

9. **Type the `root` password for the remote system.**

   You can now remotely administer print queues on the remote system in the same way that you administer local print queues.

Selecting a Print Device

When you set up a new printer or when you modify the properties of a configured printer, you must select an appropriate device for that printer. The following table describes the device choices that might be displayed in the Select Device window.

Device	Description	When to Use
printer-name	Specifies a printer that has been automatically detected.	Select this device when setting up a new printer.
printer-name (*serial-number*)	Specifies a printer that has been automatically detected, and includes the serial number.	Select this device when setting up a newly detected printer. **Note –** Most often, this device is the same device as the *printer-name* device. The difference is that one entry contains the serial number of the printer, and the other entry does not. The reason two entries are listed for the same device is that both the `system-config-printer` back end and the HAL back end detect USB devices. When configuring a new printer, you can specify either device.
Serial Port # *number*	Specifies a device that is connected to the serial port of your local system.	Select this device when setting up a printer that is connected to your local system's serial port.
AppSocket/HP Jet Direct	Specifies a device that uses a method of communication with network printers that is effectively a TCP connection.	Select this device when setting up remote print queues on a network.
Internet Printing Protocol (IPP)	Specifies a device that is used for network printer setup on a host that is running IPP.	Select this device when setting up newer printer models on a host that is running IPP.
LPD/LPR Host or Printer	Specifies a device that is used to connect to an LPD network printer.	Select this device when setting up remote print queues that use LPD. **Note –** This device might not work for more modern printer models.

Device	Description	When to Use
Unknown	Specifies a device that uses a Server Message Block (SMB) host on the network.	Select this device when setting up printers on Windows-hosted systems. **Note –** This device might not be displayed on all systems.
Other	Specifies a device that uses a device URI that is user-defined.	Select this device when setting up printers by specifying your own destination or device URI, for example, `file:///dev/printers/0`. Note that `file: device uri` support must be enabled under CUPS (`cupsctl FileDevice=yes`).

How to Set Up a New Local Printer

The following procedure describes how to set up a new locally attached printer by using the CUPS Print Manager GUI.

1. **Connect the new printer to your local system, then power it on.**

   - **When the printer is detected by the system, the Printer configuration dialog appears, displaying information about the newly detected printer.**
   - **If you are adding a new printer that was not automatically detected, do the following:**

     a. **Start CUPS Print Manager by choosing System → Administration → Print Manager from the desktop's main menubar or by typing the following command in a terminal window:**

     ```
     $ system-config-printer
     ```

     b. **Choose Server → New → Printer from the main menu.**

     Alternatively, you can click the New icon that is located on the menubar.

     c. **When prompted, type the `root` password.**

     The Printer configuration dialog appears, displaying all of the configured printers and the newly connected printer.

2. **In the Select Device window, select the appropriate device, then click Forward.**

   By default, CUPS selects the USB device that is physically connected to your system or the device that was detected by HAL. Note that these two entries might be for the same printer. For more information about selecting a device, see "Selecting a Print Device" on page 280.

3. **In the Choose Driver window, select a make for your printer, then click Forward.**

4. **Determine whether to accept the default printer driver or provide a PPD file.**

   - **To use the default driver, leave the Select Printer From Database option selected.**
   - **To provide a PPD file:**

     a. **Select the Provide PPD File option.**
        The Select a File window is displayed.

     b. **Locate the specified PPD file on your system, then click Open to associate the PPD file with the new printer.**

5. **From the left pane of the next Choose Driver window, select a printer model. From the right pane, select a printer driver. Then, click Forward.**
   By default, CUPS selects a "recommended" printer model and driver for your printer. However, you can optionally make another selection from the list of available drivers.

6. **In the Installable Options window, change any of the options that are available for your particular printer model, then click Forward.**
   For more information, see "Configurable Printer Properties" on page 283.

7. **In the Describe Printer window, provide the following information:**

   - Printer Name
   - Description
   - Location

8. **To save your changes, click Apply. If prompted, type the `root` password.**
   After you have saved your changes, the newly configured printer is displayed in the CUPS Print Manager window.

9. **(Optional) To set the printer as the default, right-click the printer name.**

   a. **Choose the Set as Default option.**

   b. **In the Set Default Printer window, choose one of the following options:**

      - Set as the system-wide default printer (default)
      - Set as my personal default printer

10. **Click OK to save the printer configuration.**

11 **(Optional) To verify that the printer is configured correctly and is working, print a test page.**

Administering Printers by Using CUPS Print Manager (Task Map)

Task	Description	For Information
Modify the properties of a configured printer.	Use CUPS Print Manager to view or change settings for a configured printer.	"How to Modify the Properties of a Configured Printer" on page 285
Rename or copy a printer configuration.	Use CUPS Print Manager to rename printers or copy an existing printer's configuration.	"How to Rename a Printer" on page 286
Delete an existing printer.	Use CUPS Print Manager to delete a configured printer.	"How to Delete a Printer" on page 287
Unshare or share a printer.	Use CUPS Print Manager to unshare or share a printer.	"How to Unshare or Share a Printer" on page 287
Disable or enable a printer.	Use CUPS Print Manager to disable or enable a printer.	"How to Disable or Enable a Printer" on page 288
Manage print jobs for configured printers.	Use CUPS Print Manager to view and manage print jobs for configured printers.	"How to Manage Print Jobs for a Specified Printer" on page 288

Administering Printers by Using CUPS Print Manager

This section describes how to administer printers by using CUPS Print Manager.

Configurable Printer Properties

Use the options in the Printer Properties dialog to modify the properties of a configured printer. For instructions, see "How to Modify the Properties of a Configured Printer" on page 285.

The Printer Properties dialog includes the following six sections for configuring new and existing printers:

- **Settings**

  In the Settings section, you can configure the following properties:

  Description Descriptive text about the printer.

Location	A description of the physical location of the printer.
Device URI	Information about the protocol that is used to access the printer. For example, you could use LPD to specify the RFC-1179 protocol or IPP to specify the Internet Printing Protocol.
Make and Model	Information about the make and model of the printer. The default setting for the Make and Model option can be changed by clicking the Change button.
Printer State	Information about the current status of the printer.
Tests and Maintenance	Contains the following options: ■ Print Test Page ■ Print Self-Test Page ■ Clean Print Heads

- **Policies**

  In the Policies section, you can configure the properties that control how a printer behaves.

State	Specifies the following printer states: ■ Enabled ■ Accepting Requests ■ Shared Note that more than one state can be specified at the same time.
Policies	Specifies how the printer behaves during error conditions.
Banner	Specifies whether starting or ending banner pages are printed with each print job.

- **Access Control**

  The Allow or Deny lists determine which users can print to the printer.

- **Printer Options**

  In the Printer Options section, you can configure printer-specific options.

  For example, for an HP LaserJet 3015, the following configurable options are displayed:

  - Imaging Options
  - Resolution Options
  - Watermark/Overlay

  The number and types of options are determined by the PPD file that is associated with the specified printer.

- **Job Options**

Determines the options that are associated with a print job, for example, the number of copies and page orientation, as well as certain image options. The number and types of options are determined by the PPD file that is associated with the specified printer.

How to Modify the Properties of a Configured Printer

The following procedure describes how to modify the basic configuration of an existing printer. For a complete description of all of the properties that you can modify by using the CUPS Print Manager GUI, see "Configurable Printer Properties" on page 283.

1. **Start the CUPS Print Manager GUI by choosing System → Administration → Print Manager from the desktop's main menubar or by typing the following command in a terminal window:**

   ```
   $ system-config-printer
   ```

   The Printer configuration dialog is displayed listing all of the configured printers and any newly detected printers.

2. **Right-click the name of the printer for which you want to modify the properties, then choose Properties.**

   The Printer Properties dialog appears. The Properties dialog contains five separate sections, each of which contains properties that are grouped by category. By default, the Settings section of the dialog is displayed.

   In the Settings section, you can modify the following settings:

   - Description
   - Location
   - Device URI
   - Make and Model

3. **To modify the printer description or location, type the new information in the corresponding text field.**

4. **To modify the device URI:**

   a. **Click the Change button next to the setting.**

   b. **From the list of available devices, select a device, then click Apply.**

      For a description of the available devices, see "Selecting a Print Device" on page 280.

   c. **When prompted, type the `root` password.**

      You are returned to the Settings section.

5 **To modify the printer make and model:**

   a. **Click the Change button next to the setting.**

   b. **In the Choose Driver window, select a printer make, then click Forward.**

      Note – By default, CUPS uses the Select Printer From Database option and selects the appropriate printer-make for you. Alternatively, you can provide your own PPD file. For instructions, see Step 3 of "How to Set Up a New Local Printer" on page 281.

   c. **From the left pane of the next Choose Driver window, select a printer model. From the right pane, select a printer driver, then click Forward.**

   d. **In the Existing Settings dialog, choose from the following options, then click Apply.**

      - Use the new PPD (Postscript Printer Description) as is.
      - Try to copy the option settings over from the old PPD.

   e. **If prompted, type the `root` password.**

      You are returned to the Settings section of the Printer Properties dialog.

6 **Click OK.**

▼ How to Rename a Printer

1 **Start the CUPS Print Manager GUI by choosing System → Administration → Print Manager from the desktop's main menubar or by typing the following command in a terminal window:**

```
$ system-config-printer
```

   The Printer configuration dialog appears, listing all of the configured printers and any newly detected printers.

2 **Right-click the name of the printer that you want to rename.**

3 **Choose the Rename option.**

4 **Type a new name for the printer.**

5 **Type the `root` password when prompted.**

6 **Click OK to save the changes.**

How to Copy a Printer Configuration

1. **Start the CUPS Print Manager GUI by choosing System → Administration → Print Manager from the desktop's main menubar or by typing the following command in a terminal window:**

   ```
   $ system-config-printer
   ```

   The Printer configuration dialog appears, listing all of the configured printers and any newly detected printers.

2. **Right-click the name of the printer that you want to copy the configuration.**

3. **Choose the Copy option.**

4. **In the Copy Printer window, type a name for the printer, then click OK.**

5. **Type the `root` password when prompted.**

6. **Click OK.**

How to Delete a Printer

1. **Start the CUPS Print Manager GUI by choosing System → Administration → Print Manager from the desktop's main menubar or by typing the following command in a terminal window:**

   ```
   $ system-config-printer
   ```

   The Printer configuration dialog appears, listing all of the configured printers and any newly detected printers.

2. **Right-click the name of the printer that you want to delete, then choose Delete.**

3. **Click OK in the Confirm Deletion dialog.**

How to Unshare or Share a Printer

By default, new printers are configured with the Share option enabled, which means they are published on the local network. This procedure describes how to unshare a printer or enable an unshared printer.

1. **Start the CUPS Print Manager GUI by choosing System → Administration → Print Manager from the desktop's main menubar or by typing the following command in a terminal window:**

   ```
   $ system-config-printer
   ```

The Printer configuration dialog appears, listing all of the configured printers and any newly detected printers.

2. **Right-click the printer name that you want to unshare, or share, then deselect the option.**
3. **Type the `root` password when prompted.**
4. **Click OK.**

How to Disable or Enable a Printer

When you configure a new printer by using CUPS Print Manager, the printer is enabled by default. This procedure describes how to disable or enable a printer.

1. **Start the CUPS Print Manager GUI by choosing System → Administration → Print Manager from the desktop's main menubar or by typing the following command in a terminal window:**

   ```
   $ system-config-printer
   ```

   The Printer configuration dialog appears, listing all of the configured printers and any newly detected printers.

2. **Right-click the name of the printer that you want to disable, or enable, then deselect the option.**
3. **Type the `root` password when prompted.**
4. **Click OK.**

How to Manage Print Jobs for a Specified Printer

1. **Start the CUPS Print Manager GUI by choosing System → Administration → Print Manager from the desktop's main menubar or by typing the following command in a terminal window:**

   ```
   $ system-config-printer
   ```

   The Printer configuration dialog appears, listing all of the configured printers and any newly detected printers.

2. **Right-click the name of the printer for which you want to manage print jobs, then choose View Print Queue.**

   The Document Print Status (*printer-name*) window appears, listing all of the print jobs for the specified printer.

In this window, you can view the following information:

- Job
- User
- Document
- Printer size
- Time submitted
- Status

3 To view information about completed jobs or printer status, select the appropriate option from the View menu.

4 To perform a specific action on a print job, select the print job, then select an action from the available choices on the menubar.

Alternatively, you can right-click the name of a print job, and from the list of available options, select an action.

You can view the following actions:

- Cancel
- Hold
- Release
- Reprint

5 (Optional) To refresh the View Print Queue window, choose View → Refresh.

◆ ◆ ◆ CHAPTER 16

Managing the System Console, Terminal Devices, and Power Services (Tasks)

This chapter describes how to manage the system console and locally connected terminal devices through the `ttymon` program and system power services.

This is a list of the information that is in this chapter:

- "What's New in Managing the System Console and Locally Connected Terminal Devices" on page 291
- "Managing the System Console and Locally Attached Connected Terminal Devices (Task Map)" on page 293
- "Overview of the System Console and Locally Connected Terminal Devices" on page 293
- "Managing the System Console and Locally Connected Terminal Devices" on page 294
- "Managing System Power Services" on page 296

What's New in Managing the System Console and Locally Connected Terminal Devices

The following features are new or changed in Oracle Solaris 11.

Removal of Support for SVR4 Service Access Facility Commands and Service Access Controller Program

The `sac` command and the Service Access Facility (SAF) program are not supported in Oracle Solaris 11.

If you want to offer login services on auxiliary terminals, you can use one of the following services:

- `svc:/system/console-login:terma`
- `svc:/system/console-login:termb`

Alternatively, you can create your own instances of the `console-login` service by creating service profiles or by using separate service manifests. For more information, see Chapter 6, "Managing Services (Overview)."

Virtual Terminal Support

The virtual console, also known as the virtual terminal (VT) device driver, provides management functions that enable you to switch between multiple screens on a single physical device. VTs are accessed the same way as any other device on the system. VTs provide the link between different screen faces and a device. The virtual console that corresponds to the currently visible screen face is the *active virtual console*. In Oracle Solaris 11, the SMF service that manages VT functionality is enabled by default.

Besides the system console, which runs on `/dev/console`, and Xorg which uses the seventh virtual console (`/dev/vt/7`), there are five login prompts for virtual console instances:

```
# svcs  | grep login
online         17:49:11 svc:/system/console-login:default
online         17:49:11 svc:/system/console-login:vt2
online         17:49:11 svc:/system/console-login:vt3
online         17:49:11 svc:/system/console-login:vt4
online         17:49:11 svc:/system/console-login:vt5
online         17:49:11 svc:/system/console-login:vt6
```

To switch between virtual console terminals, use the Alt + Ctrl + F# hotkey combination. For example, to use vt2, press Alt + Ctrl + F2. You can also create graphical VT sessions and then switch between those sessions by using the User Switcher panel applet in the desktop. To add the applet to the desktop, right click the panel, then select the Add to Panel... option. To switch to a new or different graphical login session, click the applet, then select Switch User.

To enable, disable, and modify properties for virtual consoles, as well as add and remove virtual consoles, use the `svccfg` command. For more information and examples, see the `vtdaemon(1M)` man page.

Bitmapped Console Support

Oracle Solaris 11 supports higher resolution and color depth on x86 based systems than the older Video Graphics Array (VGA) 640-480 16-color console. This support is provided for systems that use traditional BIOS and Video Electronics Standards Association (VESA) option read-only memory (ROM). Note that support is limited to when a graphics card or frame buffer is used as a physical or virtual console. There is no impact on the behavior of serial consoles. For more information, see "Support for Bitmapped Console" in *Booting and Shutting Down Oracle Solaris on x86 Platforms*.

Managing the System Console and Locally Attached Connected Terminal Devices (Task Map)

Task	Description	For Instructions
Modify settings for the system console.	Run the `svccfg` command to set the property for the service instance that you want to change.	"How to Modify Settings for the System Console" on page 294
Set up login services on auxiliary terminals.	To set up login services on auxiliary terminals, use one of the following services: ■ `svc:/system/console-login-terma` ■ `svc:system/console-login-termb`	"How to Set Up Login Services on Auxiliary Terminals" on page 295
Modify console and terminal settings by using the `eeprom` command.	You can modify console terminal settings, for example, the baud rate speed, by using the `eeprom` command.	"How to Set the Baud Rate Speed on the System Terminal" on page 295

Overview of the System Console and Locally Connected Terminal Devices

The system console is a terminal that has special attributes and is used for certain purposes. For example. kernel messages that are meant for an administrator are sent to the Console and not other terminals.

A terminal is a means of interacting with Oracle Solaris. Your system's bitmapped graphics display is not the same as an alphanumeric terminal. An alphanumeric terminal connects to a serial port and displays only text. You do not have to perform any special steps to administer the graphics display.

A terminal could also be associated with the physical monitor and keyboard layout of a computer. What sets the graphical terminal apart is that it must be associated with the graphics card and monitor of a computer. So, instead of transmitting characters out of a serial port, the characters are drawn onto the memory of the graphics card that is in the computer.

SMF Services That Manage the System Console and Locally Connected Terminal Devices

The system console and locally connected terminal devices are represented as instances of the SMF service, `svc:/system/console`. This service defines most of the behavior, with each instance having specific overrides to the settings that are inherited from the service. The `ttymon`

program is used to offer login services for these terminals. Each terminal uses a separate instance of the `ttymon` program. Command-line arguments that are passed by the service to the `ttymon` program govern its behavior.

The service instances that are supplied with the system are as follows:

- `svc:/system/console-login:default`

  The default instance always represents that the `ttymon` program offer a login to the system hardware console. For an example, see "How to Modify Settings for the System Console" on page 294.

- `svc:/system/console-login:{vt2, vt3, vt4, vt5, vt6}`

  Additional service instances are provided for the system's virtual consoles. If virtual consoles are not available, these services are automatically disabled. For more information, see the `vtdaemon(1M)` man page.

- `svc:/system/console-login:{terma, termb}`

  The `svc:/system/console-login:terma` and `svc:/system/console-login:termb` services are provided as a convenience. These services an assist you in setting up login services for additional `/dev/term/a` and `/dev/term/b` ports. These services are *disabled* by default.

You can define additional service instances as part of the svc:system/console-login service. For example, if you had a /dev/term/f device which you needed to support, you could instantiate 'svc:/system/console-login:termf' and configure it appropriately.

Managing the System Console and Locally Connected Terminal Devices

Administration of the system console is managed by SMF. Use the `svccfg` command to set the system console properties.

▼ How to Modify Settings for the System Console

This procedure shows how to change the console terminal type by using the `svccfg` command.

1 Become the `root` role.

```
$ su -
Password:
#
```

2. **Use the `svccfg` command to set the property for the service instance that you want to change.**

   For example, to change the terminal type for the system console, which is represented by the `:default` service, you would type the following command:

   ```
   # svccfg -s svc:/system/console-login:default "setprop ttymon/terminal_type = xterm"
   ```

Caution – It is not advisable to set the terminal type of the `svc:/system/console-login` service because the change will affect *all* instances.

How to Set Up Login Services on Auxiliary Terminals

For terminals that are connected to `/dev/term/a` or `/dev/term/b` serial ports on a system, predefined services are provided.

To enable login services for `/dev/term/a`, use the following procedure.

1. **Become the `root` role.**

2. **Enable the service instance as follows:**

   ```
   # svcadm enable svc:/system/console-login:terma
   ```

3. **Check that the service is online.**

   ```
   # svcs svc:/system/console-login:terma
   ```

   The output should show that the service is online. If the service is in maintenance mode, consult the service's log file for further details.

How to Set the Baud Rate Speed on the System Terminal

This procedure shows how to set the baud rate speed on the console. Support for console speeds on x86 based systems are dependent on the specific platform.

The following are supported console speeds for SPARC based systems:

- `9600 bps`
- `19200 bps`
- `38400 bps`

1. **Become an administrator.**

2. **Use the `eeprom` command to set a baud rate speed that is appropriate for your system type.**

   ```
   # eeprom ttya-mode=baud-rate,8,n,1,-
   ```

For example, to change the baud rate on an x86 based system's console to 38400, type:

```
# eeprom ttya-mode=38400,8,n,1,-
```

3 **Change the console line in the /etc/ttydefs file as follows:**

`console` *baud-rate* `hupcl opost onlcr:`*baud-rate*`::console`

4 **Make the following additional changes for your system type.**

Note that these changes are platform-dependent.

- **On SPARC based systems:** Change the baud rate speed in the version of the `options.conf` file that is in the `/etc/driver/drv` directory.

  Use the following command to change the baud rate to 9600:

  ```
  # 9600              :bd:
  ttymodes="2502:1805:bd:8a3b:3:1c:7f:15:4:0:0:0:11:13:1a:19:12:f:17:16";
  ```

  Use the following command to change the baud rate speed to 19200.

  ```
  # 19200             :be:
  ttymodes="2502:1805:be:8a3b:3:1c:7f:15:4:0:0:0:11:13:1a:19:12:f:17:16";
  ```

  Use the following command to change the baud rate speed to 38400:

  ```
  # 38400             :bf:
  ttymodes="2502:1805:bf:8a3b:3:1c:7f:15:4:0:0:0:11:13:1a:19:12:f:17:16";
  ```

- **On x86 based systems:** Change the console speed if the BIOS serial redirection is enabled.

Managing System Power Services

In Oracle Solaris 11 release, power management configuration has moved into an SMF configuration repository. The new `poweradm` command is used to manage system power management properties directly rather than using a combination of power-related command, daemon, and configuration file. These changes are part of a wider set of changes to modernize the power management framework in Oracle Solaris 11.

The following power management features are no longer available:

- `/etc/power.conf`
- `pmconfig` and `powerd`
- Device power management

Two new properties describe the power configuration that manage time components:

- `time-to-full-capacity` – Defines the maximum time the system is allowed to reach its full capacity, from any lower-capacity or less-responsive state, while the system is active.
- `time-to-minimum-responsiveness` – Defines how long the system is allowed to return to its active state.

You can display `poweradm` properties by using the following command:

```
# poweradm list
active_control/administrative-authority      smf=platform, current=platform
suspend/suspend-enable                       smf=false, current=false
active_config/time-to-full-capacity          platform=250, current=250
active_config/time-to-minimum-responsiveness platform=0, current=0
disabled                                     platform=false
```

In the above output, the `active_control/administrative-authority` indicates the source of the configuration with two settings:

- `platform` – Configuration for power management comes from the platform. This is the default value.
- `smf` – Allows the other power management properties to be set using the `poweradm` command.

If you previously enabled S3-support in the `/etc/power.conf` file to suspend and resume your system, similar `poweradm` syntax is:

```
# poweradm set suspend-enable=true
```

The `suspend-enable` property is set to false by default.

Use the following syntax to disable power management:

```
# poweradm set administrative-authority=none
```

Disabling the following SMF power management service does not disable power management:

```
online         Sep_02   svc:/system/power:default
```

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

How to Recover from Power Service in Maintenance Mode

If `administrative-authority` is set to `smf` before both time-to-full-capacity and time-to-minimum-responsiveness have been set, the service will go into maintenance mode.

1. **Become an administrator.**
2. **Set `administrative-authority` to `none`.**

   ```
   # poweradm set administrative-authority=none
   ```

3. **Set both `time-to-full-capacity` and `time-to-minimum-responsiveness` to their desired values.**

   ```
   # poweradm set time-to-full-capacity=value
   # poweradm set time-to-minimum-responsiveness=value
   ```

4. **Clear the service.**

   ```
   # svcadm clear power
   ```

5. **Set `administrative-authority` to `smf`.**

   ```
   # poweradm set administrative-authority=smf
   ```

◆ ◆ ◆ CHAPTER 17

Managing System Crash Information (Tasks)

This chapter describes how to manage system crash information in the Oracle Solaris OS.

This is a list of the information that is in this chapter:

- "What's New in Managing System Crash Information" on page 299
- "Managing System Crash Information (Task Map)" on page 300
- "System Crashes (Overview)" on page 300
- "Managing System Crash Dump Information" on page 303

What's New in Managing System Crash Information

This section describes new or changed features for managing system resources in this Oracle Solaris release.

Fast Crash Dump Facility

This feature enhancement enables the system to save crash dumps in less time, using less space. The time that is required for a crash dump to complete is now two to ten times faster, depending on the platform. The amount of disk space that is required to save crash dumps in the `savecore` directory is reduced by the same factors. To accelerate the creation and compression of the crash dump file, the fast crash dump facility utilizes lightly used CPUs on large systems. A new crash dump file, `vmdump.`*n*, is a compressed version of the `vmcore.`*n* and `unix.`*n* files. Compressed crash dumps can be moved over the network more quickly and then analyzed off-site. Note that the dump file must first be uncompressed to use it with tools like the `mdb` utility. You can uncompress a dump file by using the `savecore` command, either locally or remotely.

To support the new crash dump facility, the `-z` option has been added to the `dumpadm` command. Use this option to specify whether to save dumps in a compressed or an uncompressed format. The default format is compressed.

For more detailed information, see the dumpadm(1M) and the savecore(1M) man pages.

Managing System Crash Information (Task Map)

Task	Description	For Instructions
1. Display the current crash dump configuration.	Display the current crash dump configuration by using the dumpadm command.	"How to Display the Current Crash Dump Configuration" on page 303
2. Modify the crash dump configuration.	Use the dumpadm command to specify the type of data to dump, whether or not the system will use a dedicated dump device, the directory for saving crash dump files, and the amount of space that must remain available after crash dump files are written.	"How to Modify a Crash Dump Configuration" on page 304
3. Examine a crash dump file.	Use the mdb command to view crash dump files.	"How to Examine a Crash Dump" on page 305
4. (Optional) Recover from a full crash dump directory.	The system crashes, but no room is available in the savecore directory, and you want to save some critical system crash dump information.	"How to Recover From a Full Crash Dump Directory (Optional)" on page 306
5. (Optional) Disable or enable the saving of crash dump files.	Use the dumpadm command to disable or enable the saving the crash dump files. Saving of crash dump files is enabled by default.	"How to Disable or Enable the Saving of Crash Dumps" on page 307

System Crashes (Overview)

System crashes can occur due to hardware malfunctions, I/O problems, and software errors. If the system crashes, it will display an error message on the console, and then write a copy of its physical memory to the dump device. The system will then reboot automatically. When the system reboots, the savecore command is executed to retrieve the data from the dump device and write the saved crash dump to your savecore directory. The saved crash dump files provide invaluable information to your support provider to aid in diagnosing the problem.

The crash dump information is written in a compressed format to the vmdump.*n* file, where n is an integer that identifies the crash dump. Afterwards, the savecore command can be invoked on the same system or another system to expand the compressed crash dump to a pair of files that are named unix.*n* and vmcore.*n*. The directory in which the crash dump is saved upon reboot can also be configured by using the dumpadm command.

For systems that have an Oracle Solaris ZFS root file system, dedicated ZFS volumes are used for swap and dump areas. See "Managing Your ZFS Swap and Dump Devices" in *Oracle Solaris Administration: ZFS File Systems* for more information.

x86: System Crashes in the GRUB Boot Environment

If a system crash occurs on an x86 based system in the GRUB boot environment, it is possible that the SMF service that manages the GRUB boot archive, `svc:/system/boot-archive:default`, might fail on the next system reboot. For more information about GRUB based booting, see *Booting and Shutting Down Oracle Solaris on x86 Platforms*.

System Crash Dump Files

The `savecore` command runs automatically after a system crash to retrieve the crash dump information from the dump device and writes a pair of files called `unix.`*X* and `vmcore.`*X*, where *X* identifies the dump sequence number. Together, these files represent the saved system crash dump information.

Crash dump files are sometimes confused with *core* files, which are images of user applications that are written when the application terminates abnormally.

Crash dump files are saved in a predetermined directory, which by default, is `/var/crash/`. In previous releases, crash dump files were overwritten when a system rebooted, unless you manually enabled the system to save the images of physical memory in a crash dump file. Now, the saving of crash dump files is enabled by default.

System crash information is managed with the `dumpadm` command. For more information, see "The `dumpadm` Command" on page 302.

Saving Crash Dumps

You can examine the control structures, active tables, memory images of a live or crashed system kernel, and other information about the operation of the kernel by using the `mdb` utility. Using `mdb` to its full potential requires a detailed knowledge of the kernel, and is beyond the scope of this manual. For information about using this utility, see the `mdb(1)` man page.

Additionally, crash dumps saved by `savecore` can be useful to send to a customer service representative for analysis of why the system is crashing.

The dumpadm Command

Use the dumpadm command to manage system crash dump information in the Oracle Solaris OS.

- The dumpadm command enables you to configure crash dumps of the operating system. The dumpadm configuration parameters include the dump content, dump device, and the directory in which crash dump files are saved.
- Dump data is stored in compressed format on the dump device. Kernel crash dump images can be as big as 4 Gbytes or more. Compressing the data means faster dumping and less disk space needed for the dump device.
- Saving crash dump files is run in the background when a dedicated dump device, not the swap area, is part of the dump configuration. This means a booting system does not wait for the savecore command to complete before going to the next step. On large memory systems, the system can be available before savecore completes.
- System crash dump files, generated by the savecore command, are saved by default.
- The savecore -L command is a new feature which enables you to get a crash dump of the live running the Oracle Solaris OS. This command is intended for troubleshooting a running system by taking a snapshot of memory during some bad state, such as a transient performance problem or service outage. If the system is up and you can still run some commands, you can execute the savecore -L command to save a snapshot of the system to the dump device, and then immediately write out the crash dump files to your savecore directory. Because the system is still running, you can only use the savecore -L command if you have configured a dedicated dump device.

The following table describes dumpadm's configuration parameters.

Dump Parameter	Description
dump device	The device that stores dump data temporarily as the system crashes. When the dump device is not the swap area, savecore runs in the background, which speeds up the boot process.
savecore directory	The directory that stores system crash dump files.
dump content	Type of memory data to dump.
minimum free space	Minimum amount of free space required in the savecore directory after saving crash dump files. If no minimum free space has been configured, the default is one Mbyte.

For more information, see dumpadm(1M).

Dump configuration parameters are managed by the dumpadm command.

How the dumpadm Command Works

During system startup, the dumpadm command is invoked by the svc:/system/dumpadm:default service to configure crash dumps parameters.

Specifically, dumpadm initializes the dump device and the dump content through the /dev/dump interface.

After the dump configuration is complete, the savecore script looks for the location of the crash dump file directory. Then, savecore is invoked to check for crash dumps and check the content of the minfree file in the crash dump directory.

Managing System Crash Dump Information

Keep the following key points in mind when you are working with system crash information:

- You must be the root user to access and manage system crash information.
- Do not disable the option of saving system crash dumps. System crash dump files provide an invaluable way to determine what is causing the system to crash.
- Do not remove important system crash information until it has been sent to your customer service representative.

▼ How to Display the Current Crash Dump Configuration

1. **Become the root role.**

2. **Display the current crash dump configuration.**

```
# dumpadm
Dump content: kernel pages
Dump device: /dev/dsk/c0t3d0s1 (swap)
Savecore directory: /var/crash
 Savecore enabled: yes
  Saved compressed: on
```

The preceding example output means:

- The dump content is kernel memory pages.
- Kernel memory will be dumped on a swap device, /dev/dsk/c0t3d0s1. You can identify all your swap areas with the swap -l command.
- System crash dump files will be written in the /var/crash directory.
- Saving crash dump files is enabled.
- Save crash dumps in compressed format.

▼ How to Modify a Crash Dump Configuration

1 Become the `root` role.

2 Identify the current crash dump configuration.

```
# dumpadm
      Dump content: kernel pages
       Dump device: /dev/dsk/c0t3d0s1 (swap)
Savecore directory: /var/crash
 Savecore enabled: yes
     Save commpressed: on
```

This output identifies the default dump configuration for a system running the Oracle Solaris 10 release.

3 Modify the crash dump configuration.

```
#  /usr/sbin/dumpadm  [-nuy] [-c content-type] [-d dump-device] [-m mink | minm | min%]
[-s savecore-dir] [-r root-dir] [-z on | off]
```

`-c` *content*	Specifies the type of data to dump. Use `kernel` to dump of all kernel memory, `all` to dump all of memory, or `curproc`, to dump kernel memory and the memory pages of the process whose thread was executing when the crash occurred. The default dump content is kernel memory.
`-d` *dump-device*	Specifies the device that stores dump data temporarily as the system crashes. The primary swap device is the default dump device.
`-m` *nnnk* | *nnnm* | *nnn%*	Specifies the minimum free disk space for saving crash dump files by creating a `minfree` file in the current `savecore` directory. This parameter can be specified in Kbytes (`nnnk`), Mbytes (`nnnm`) or file system size percentage (`nnn%`). The `savecore` command consults this file prior to writing the crash dump files. If writing the crash dump files, based on their size, would decrease the amount of free space below the `minfree` threshold, the dump files are not written and an error message is logged. For information about recovering from this scenario, see "How to Recover From a Full Crash Dump Directory (Optional)" on page 306.
`-n`	Specifies that `savecore` should not be run when the system reboots. This dump configuration is not recommended. If system crash information is written to the swap device, and `savecore` is not enabled, the crash dump information is overwritten when the system begins to swap.
`-s`	Specifies an alternate directory for storing crash dump files. In Oracle Solaris 11, the default directory is `/var/crash`.

`-u`	Forcibly updates the kernel dump configuration based on the contents of the `/etc/dumpadm.conf` file.
`-y`	Modifies the dump configuration to automatically execute the `savecore` command upon reboot, which is the default for this dump setting.
`-z on | off`	Modifies the dump configuration to control the operation of the `savecore` command upon reboot. The `on` setting enables the saving of core file in a compressed format. The `off` setting automatically uncompresses the crash dump file. Because crash dump files can be extremely large and therefore require less file system space if they are saved in a compressed forma, the default is `on`.

Example 17–1 Modifying a Crash Dump Configuration

In this example, all of memory is dumped to the dedicated dump device, `/dev/dsk/c0t1d0s1`, and the minimum free space that must be available after the crash dump files are saved is 10% of the file system space.

```
# dumpadm
      Dump content: kernel pages
       Dump device: /dev/dsk/c0t3d0s1 (swap)
Savecore directory: /var/crash
  Savecore enabled: yes
   Save compressed: on
 # dumpadm -c all -d /dev/dsk/c0t1d0s1 -m 10%
      Dump content: all pages
       Dump device: /dev/dsk/c0t1d0s1 (dedicated)
Savecore directory: /var/crash (minfree = 77071KB)
  Savecore enabled: yes
    Save compressed: on
```

▼ How to Examine a Crash Dump

1 Become the `root` role.

2 Examine a crash dump by using the `mdb` utility.

`# /usr/bin/mdb [-k]` *crashdump-file*

`-k`	Specifies kernel debugging mode by assuming the file is an operating system crash dump file.
crashdump-file	Specifies the operating system crash dump file.

3 Display crash status information.

```
# /usr/bin/mdb file-name
> ::status
   .
   .
   .
> ::system
   .
   .
   .
```

Example 17–2 Examining a Crash Dump

The following example shows sample output from the `mdb` utility, which includes system information and identifies the tunables that are set in this system's `/etc/system` file.

```
# /usr/bin/mdb -k unix.0
Loading modules: [ unix krtld genunix ip nfs ipc ptm ]
> ::status
debugging crash dump /dev/mem (64-bit) from ozlo
operating system: 5.10 Generic sun4v
> ::system
set ufs_ninode=0x9c40 [0t40000]
set ncsize=0x4e20 [0t20000]
set pt_cnt=0x400 [0t1024]
```

▼ How to Recover From a Full Crash Dump Directory (Optional)

In this scenario, the system crashes but no room is left in the `savecore` directory, and you want to save some critical system crash dump information.

1 After the system reboots, log in as the `root` user.

2 Clear out the `savecore` directory, typically, `/var/crash/`, by removing existing crash dump files that have already been sent to your service provider.

- **Alternatively, you can manually run the `savecore` command to specify an alternate directory that has sufficient disk space.**

  ```
  # savecore [ directory ]
  ```

▼ How to Disable or Enable the Saving of Crash Dumps

1. **Become the `root` role.**

2. **Disable or enable the saving of crash dumps on your system.**

```
# dumpadm -n | -y
```

Example 17–3 Disabling the Saving of Crash Dumps

This example illustrates how to disable the saving of crash dumps on your system.

```
# dumpadm -n
      Dump content: all pages
       Dump device: /dev/dsk/c0t1d0s1 (dedicated)
Savecore directory: /var/crash (minfree = 77071KB)
  Savecore enabled: no
   Save Compressed: on
```

Example 17–4 Enabling the Saving of Crash Dumps

This example illustrates how to enable the saving of crash dump on your system.

```
# dumpadm -y
      Dump content: all pages
       Dump device: /dev/dsk/c0t1d0s1 (dedicated)
Savecore directory: /var/crash (minfree = 77071KB)
  Savecore enabled: yes
   Save compressed: on
```

◆ ◆ ◆ CHAPTER 18

Managing Core Files (Tasks)

This chapter describes how to manage core files with the `coreadm` command.

This is a list of the information that is in this chapter;

- "Managing Core Files (Task Map)" on page 309
- "Managing Core Files Overview" on page 310
- "Troubleshooting Core File Problems" on page 314
- "Examining Core Files" on page 314

Managing Core Files (Task Map)

Task	Description	For Instructions
1. Display the current core dump configuration.	Display the current core dump configuration by using the `coreadm` command.	"How to Display the Current Core Dump Configuration" on page 312
2. Modify the core dump configuration.	Modify the core dump configuration to do one of the following: Set a core file name pattern. Enable a per-process core file path. Enable a global core file path.	"How to Set a Core File Name Pattern" on page 313 "How to Enable a Per-Process Core File Path" on page 313 "How to Enable a Global Core File Path" on page 313
3. Examine a core dump file.	Use the `proc` tools to view a core dump file.	"Examining Core Files" on page 314

Managing Core Files Overview

Core files are generated when a process or application terminates abnormally. Core files are managed with the `coreadm` command.

For example, you can use the `coreadm` command to configure a system so that all process core files are placed in a single system directory. This means it is easier to track problems by examining the core files in a specific directory whenever a process or daemon terminates abnormally.

Configurable Core File Paths

Two new configurable `core` file paths that can be enabled or disabled independently of each other are:

- A per-process core file path, which defaults to `core` and is enabled by default. If enabled, the per-process core file path causes a `core` file to be produced when the process terminates abnormally. The per-process path is inherited by a new process from its parent process.

  When generated, a per-process core file is owned by the owner of the process with read/write permissions for the owner. Only the owning user can view this file.
- A global core file path, which defaults to `core` and is disabled by default. If enabled, an *additional* core file with the same content as the per-process core file is produced by using the global core file path.

  When generated, a global core file is owned by superuser with read/write permissions for superuser only. Non-privileged users cannot view this file.

When a process terminates abnormally, it produces a core file in the current directory by default. If the global core file path is enabled, each abnormally terminating process might produce two files, one in the current working directory, and one in the global core file location.

By default, a `setuid` process does not produce core files using either the global or per-process path.

Expanded Core File Names

If a global `core` file directory is enabled, `core` files can be distinguished from one another by using the variables described in the following table.

Variable Name	Variable Definition
`%d`	Executable file directory name, up to a maximum of `MAXPATHLEN` characters
`%f`	Executable file name, up to a maximum of `MAXCOMLEN` characters

Variable Name	Variable Definition
%g	Effective group ID
%m	Machine name (uname -m)
%n	System node name (uname -n)
%p	Process ID
%t	Decimal value of time(2)
%u	Effective user ID
%z	Name of the zone in which process is executed (zonename)
%%	Literal %

For example, if the global core file path is set to:

```
/var/core/core.%f.%p
```

and a `sendmail` process with PID `12345` terminates abnormally, it produces the following `core` file:

```
/var/core/core.sendmail.12345
```

Setting the Core File Name Pattern

You can set a core file name pattern on a global, zone, or per-process basis. In addition, you can set the per-process defaults that persist across a system reboot.

For example, the following `coreadm` command sets the default per-process core file pattern. This setting applies to all processes that have not explicitly overridden the default core file pattern. This setting persists across system reboots. For example in Solaris 9, the following `coreadm` command sets the global core file pattern for all processes started by the `init` process. This pattern will persist across system reboots.

```
# coreadm -i /var/core/core.%f.%p
```

The following `coreadm` command sets the per-process core file name pattern for any processes:

```
# coreadm -p /var/core/core.%f.%p $$
```

The `$$` symbols represent a placeholder for the process ID of the currently running shell. The per-process core file name pattern is inherited by all child processes.

Once a global or per-process core file name pattern is set, it must be enabled with the `coreadm -e` command. See the following procedures for more information.

You can set the core file name pattern for all processes run during a user's login session by putting the command in a user's `$HOME/.profile` or `.login` file.

Enabling `setuid` Programs to Produce Core Files

You can use the `coreadm` command to enable or disable `setuid` programs to produce core files for all system processes or on a per-process basis by setting the following paths:

- If the global `setuid` option is enabled, a global core file path allows all `setuid` programs on a system to produce `core` files.
- If the per-process `setuid` option is enable, a per-process core file path allows specific `setuid` processes to produce `core` files.

By default, both flags are disabled. For security reasons, the global core file path must be a full pathname, starting with a leading `/`. If superuser disables per-process core files, individual users cannot obtain core files.

The `setuid` core files are owned by superuser with read/write permissions for superuser only. Regular users cannot access them even if the process that produced the `setuid` core file was owned by an ordinary user.

For more information, see the `coreadm`(1M) man page.

How to Display the Current Core Dump Configuration

Use the `coreadm` command without any options to display the current core dump configuration.

```
$ coreadm
              global core file pattern:
     global core file content: default
       init core file pattern: core
       init core file content: default
            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
```

How to Set a Core File Name Pattern

- Determine whether you want to set a per-process or global core file and select one of the following:

  a. Set a per-process file name pattern.

     ```
     $ coreadm -p $HOME/corefiles/%f.%p $$
     ```

  b. Become the `root` role.

  c. Set a global file name pattern.

     ```
     # coreadm -g /var/corefiles/%f.%p
     ```

How to Enable a Per-Process Core File Path

1. Become the `root` role.
2. Enable a per-process core file path.

   ```
   # coreadm -e process
   ```

3. Display the current process core file path to verify the configuration.

   ```
   # coreadm $$
   1180:   /home/kryten/corefiles/%f.%p
   ```

How to Enable a Global Core File Path

1. Become the `root` role.
2. Enable a global core file path.

   ```
   # coreadm -e global -g /var/core/core.%f.%p
   ```

3. Display the current process core file path to verify the configuration.

   ```
   # coreadm
         global core file pattern: /var/core/core.%f.%p
       global core file content: default
         init core file pattern: core
         init core file content: default
              global core dumps: enabled
        per-process core dumps: enabled
       global setid core dumps: disabled
   per-process setid core dumps: disabled
       global core dump logging: disabled
   ```

Troubleshooting Core File Problems

Error Message

```
NOTICE: 'set allow_setid_core = 1' in /etc/system is obsolete
NOTICE: Use the coreadm command instead of 'allow_setid_core'
```

Cause

You have an obsolete parameter that allows `setuid` core files in your `/etc/system` file.

Solution

Remove `allow_setid_core=1` from the `/etc/system` file. Then use the `coreadm` command to enable global `setuid` core file paths.

Examining Core Files

Some of the `proc` tools have been enhanced to examine process core files, as well as live processes. The `proc` tools are utilities that can manipulate features of the `/proc` file system.

The `/usr/proc/bin/pstack`, `pmap`, `pldd`, `pflags`, and `pcred` tools can now be applied to core files by specifying the name of the core file on the command line, similar to the way you specify a process ID to these commands.

For more information about using `proc` tools to examine core files, see `proc(1)`.

EXAMPLE 18–1 Examining Core Files With `proc` Tools

```
$ ./a.out
Segmentation Fault(coredump)
$ /usr/proc/bin/pstack ./core
core './core' of 19305: ./a.out
 000108c4 main     (1, ffbef5cc, ffbef5d4, 20800, 0, 0) + 1c
 00010880 _start   (0, 0, 0, 0, 0, 0) + b8
```

◆ ◆ ◆ CHAPTER 19

Troubleshooting System and Software Problems (Tasks)

This chapter provides a general overview of troubleshooting software problems, including information about troubleshooting system crashes, managing crash dump information, and viewing and managing system messages.

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

- "Troubleshooting a System Crash" on page 315
- "Managing System Messages" on page 317
- "Troubleshooting File Access Problems" on page 326

Troubleshooting a System Crash

If a system running Oracle Solaris crashes, provide your service provider with as much information as possible, including crash dump files.

What to Do If the System Crashes

The following list describes the most important information to remember in the event of a system crash:

1. Write down the system console messages.

   If a system crashes, making it run again might seem like your most pressing concern. However, before you reboot the system, examine the console screen for messages. These messages can provide some insight about what caused the crash. Even if the system reboots automatically and the console messages have disappeared from the screen, you might be able to check these messages by viewing the system error log, the `/var/adm/messages` file. For more information about viewing system error log files, see "How to View System Messages" on page 318.

If you have frequent crashes and cannot determine the cause, gather all of the information you can from the system console or the `/var/adm/messages` files and have it ready for a customer service representative to examine. For a complete list of troubleshooting information to gather for your service provider, see "Troubleshooting a System Crash" on page 315.

If the system fails to reboot successfully after a system crash, see Chapter 20, "Troubleshooting Miscellaneous System and Software Problems (Tasks)."

2. Synchronize the disks and reboot.

```
ok sync
```

If the system fails to reboot successfully after a system crash, see Chapter 20, "Troubleshooting Miscellaneous System and Software Problems (Tasks)."

Check to see if a system crash dump was generated after the system crash. System crash dumps are saved by default. For information about crash dumps, see Chapter 17, "Managing System Crash Information (Tasks)."

Gathering Troubleshooting Data

Answer the following questions to help isolate the system problem. Use "Troubleshooting a System Crash Checklist" on page 317 for gathering troubleshooting data for a crashed system.

TABLE 19–1 Identifying System Crash Data

Question	Description
Can you reproduce the problem?	This is important because a reproducible test case is often essential for debugging really hard problems. By reproducing the problem, the service provider can build kernels with special instrumentation to trigger, diagnose, and fix the bug.
Are you using any third-party drivers?	Drivers run in the same address space as the kernel, with all the same privileges, so they can cause system crashes if they have bugs.
What was the system doing just before it crashed?	If the system was doing anything unusual like running a new stress test or experiencing higher-than-usual load, that might have led to the crash.
Were there any unusual console messages right before the crash?	Sometimes the system will show signs of distress before it actually crashes; this information is often useful.
Did you add any tuning parameters to the `/etc/system` *file*?	Sometimes tuning parameters, such as increasing shared memory segments so that the system tries to allocate more than it has, can cause the system to crash.

TABLE 19–1 Identifying System Crash Data *(Continued)*

Question	Description
Did the problem start recently?	If so, did the onset of problems coincide with any changes to the system, for example, new drivers, new software, different workload, CPU upgrade, or a memory upgrade.

Troubleshooting a System Crash Checklist

Use this checklist when gathering system data for a crashed system.

Item	Your Data
Is a system crash dump available?	
Identify the operating system release and appropriate software application release levels.	
Identify system hardware.	
Include `prtdiag` output for SPARC systems. Include Explorer output for other systems.	
Are patches installed? If so, include `showrev -p` output.	
Is the problem reproducible?	
Does the system have any third-party drivers?	
What was the system doing before it crashed?	
Were there any unusual console messages right before the system crashed?	
Did you add any parameters to the `/etc/system` file?	
Did the problem start recently?	

Managing System Messages

The following sections describe system messaging features in Oracle Solaris.

Viewing System Messages

System messages display on the console device. The text of most system messages look like this:

[ID *msgid facility.priority*]

For example:

```
[ID 672855 kern.notice] syncing file systems...
```

If the message originated in the kernel, the kernel module name is displayed. For example:

```
Oct 1 14:07:24 mars ufs: [ID 845546 kern.notice] alloc: /: file system full
```

When a system crashes, it might display a message on the system console like this:

`panic:` *error message*

Less frequently, this message might be displayed instead of the panic message:

```
Watchdog reset !
```

The error logging daemon, `syslogd`, automatically records various system warnings and errors in message files. By default, many of these system messages are displayed on the system console and are stored in the `/var/adm` directory. You can direct where these messages are stored by setting up system message logging. For more information, see "Customizing System Message Logging" on page 320. These messages can alert you to system problems, such as a device that is about to fail.

The `/var/adm` directory contains several message files. The most recent messages are in `/var/adm/messages` file (and in `messages.*`), and the oldest are in the `messages.3` file. After a period of time (usually every ten days), a new `messages` file is created. The `messages.0` file is renamed `messages.1`, `messages.1` is renamed `messages.2`, and `messages.2` is renamed `messages.3`. The current `/var/adm/messages.3` file is deleted.

Because the `/var/adm` directory stores large files containing messages, crash dumps, and other data, this directory can consume lots of disk space. To keep the `/var/adm` directory from growing too large, and to ensure that future crash dumps can be saved, you should remove unneeded files periodically. You can automate this task by using the `crontab` file. For more information about automating this task, see How to Delete Crash Dump Files and Chapter 14, "Scheduling System Tasks (Tasks)."

▼ How to View System Messages

- **Display recent messages generated by a system crash or reboot by using the `dmesg` command.**

  ```
  $ dmesg
  ```

Or, use the `more` command to display one screen of messages at a time.

```
$ more /var/adm/messages
```

Example 19–1 Viewing System Messages

The following example shows output from the `dmesg` command on an Oracle Solaris 10 system.

```
$ dmesg
Mon Sep 13 14:33:04 MDT 2010
Sep 13 11:06:16 sr1-ubrm-41 svc.startd[7]: [ID 122153 daemon.warning] ...
Sep 13 11:12:55 sr1-ubrm-41 last message repeated 398 times
Sep 13 11:12:56 sr1-ubrm-41 svc.startd[7]: [ID 122153 daemon.warning] ...
Sep 13 11:15:16 sr1-ubrm-41 last message repeated 139 times
Sep 13 11:15:16 sr1-ubrm-41 xscreensaver[25520]:  ,,,
Sep 13 11:15:16 sr1-ubrm-41 xscreensaver[25520]: ...
Sep 13 11:15:17 sr1-ubrm-41 svc.startd[7]: [ID 122153 daemon.warning]...
.
.
.
```

See Also For more information, see the `dmesg(1M)` man page.

System Log Rotation

System log files are rotated by the `logadm` command from an entry in the root `crontab` file. The `/usr/lib/newsyslog` script is no longer used.

The system log rotation is defined in the `/etc/logadm.conf` file. This file includes log rotation entries for processes such as `syslogd`. For example, one entry in the `/etc/logadm.conf` file specifies that the `/var/log/syslog` file is rotated weekly unless the file is empty. The most recent `syslog` file becomes `syslog.0`, the next most recent becomes `syslog.1`, and so on. Eight previous `syslog` log files are kept.

The `/etc/logadm.conf` file also contains time stamps of when the last log rotation occurred.

You can use the `logadm` command to customize system logging and to add additional logging in the `/etc/logadm.conf` file as needed.

For example, to rotate the Apache access and error logs, use the following commands:

```
# logadm -w /var/apache/logs/access_log -s 100m
# logadm -w /var/apache/logs/error_log -s 10m
```

In this example, the Apache `access_log` file is rotated when it reaches 100 MB in size, with a `.0`, `.1`, (and so on) suffix, keeping 10 copies of the old `access_log` file. The `error_log` is rotated when it reaches 10 MB in size with the same suffixes and number of copies as the `access_log` file.

The `/etc/logadm.conf` entries for the preceding Apache log rotation examples look similar to the following:

```
# cat /etc/logadm.conf
.
.
.
/var/apache/logs/error_log -s 10m
/var/apache/logs/access_log -s 100m
```

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

You can use the `logadm` command as superuser or by assuming an equivalent role (with Log Management rights). With RBAC, you can grant non-root users the privilege of maintaining log files by providing access to the `logadm` command.

For example, add the following entry to the `/etc/user_attr` file to grant user `andy` the ability to use the `logadm` command:

```
andy::::profiles=Log Management
```

Customizing System Message Logging

You can capture additional error messages that are generated by various system processes by modifying the `/etc/syslog.conf` file. By default, the `/etc/syslog.conf` file directs many system process messages to the `/var/adm/messages` files. Crash and boot messages are stored here as well. To view `/var/adm` messages, see "How to View System Messages" on page 318.

The `/etc/syslog.conf` file has two columns separated by tabs:

facility.level ... action

facility.level — A *facility* or system source of the message or condition. May be a comma-separated listed of facilities. Facility values are listed in Table 19–2. A *level*, indicates the severity or priority of the condition being logged. Priority levels are listed in Table 19–3.

Do not put two entries for the same facility on the same line, if the entries are for different priorities. Putting a priority in the syslog file indicates that all messages of that all messages of that priority or higher are logged, with the last message taking precedence. For a given facility and level, `syslogd` matches all messages for that level and all higher levels.

action — The action field indicates where the messages are forwarded.

The following example shows sample lines from a default `/etc/syslog.conf` file.

```
user.err                                /dev/sysmsg
user.err                                /var/adm/messages
user.alert                              'root, operator'
user.emerg                              *
```

This means the following user messages are automatically logged:

- User errors are printed to the console and also are logged to the `/var/adm/messages` file.
- User messages requiring immediate action (`alert`) are sent to the `root` and `operator` users.
- User emergency messages are sent to individual users.

Note – Placing entries on separate lines might cause messages to be logged out of order if a log target is specified more than once in the `/etc/syslog.conf` file. Note that you can specify multiple selectors in a single line entry, each separated by a semicolon.

The most common error condition sources are shown in the following table. The most common priorities are shown in Table 19–3 in order of severity.

TABLE 19–2 Source Facilities for `syslog.conf` Messages

Source	Description
`kern`	The kernel
`auth`	Authentication
`daemon`	All daemons
`mail`	Mail system
`lp`	Spooling system
`user`	User processes

Note – The number of `syslog` facilities that can be activated in the `/etc/syslog.conf` file is unlimited.

TABLE 19–3 Priority Levels for `syslog.conf` Messages

Priority	Description
`emerg`	System emergencies
`alert`	Errors requiring immediate correction
`crit`	Critical errors

TABLE 19–3 Priority Levels for `syslog.conf` Messages *(Continued)*

Priority	Description
`err`	Other errors
`info`	Informational messages
`debug`	Output used for debugging
`none`	This setting doesn't log output

▼ How to Customize System Message Logging

1. **Become the `root` role.**
2. **Edit the `/etc/syslog.conf` file, adding or changing message sources, priorities, and message locations according to the syntax described in `syslog.conf`(4).**
3. **Exit the file, saving the changes.**

Example 19–2 Customizing System Message Logging

This sample `/etc/syslog.conf` `user.emerg` facility sends user emergency messages to the root user *and* individual users.

```
user.emerg                                      'root, *'
```

Enabling Remote Console Messaging

The following new console features improve your ability to troubleshoot remote systems:

- The `consadm` command enables you to select a serial device as an *auxiliary* (or remote) console. Using the `consadm` command, a system administrator can configure one or more serial ports to display redirected console messages and to host `sulogin` sessions when the system transitions between run levels. This feature enables you to dial in to a serial port with a modem to monitor console messages and participate in `init` state transitions. (For more information, see `sulogin`(1M) and the step-by-step procedures that follow.)

  While you can log in to a system using a port configured as an auxiliary console, it is primarily an output device displaying information that is also displayed on the default console. If boot scripts or other applications read and write to and from the default console, the write output displays on all the auxiliary consoles, but the input is only read from the default console. (For more information about using the `consadm` command during an interactive login session, see "Using the `consadm` Command During an Interactive Login Session" on page 324.)

- Console output now consists of kernel and `syslog` messages written to a new pseudo device, `/dev/sysmsg`. In addition, `rc` script startup messages are written to `/dev/msglog`. Previously, all of these messages were written to `/dev/console`.

  Scripts that direct console output to `/dev/console` need to be changed to `/dev/msglog` if you want to see script messages displayed on the auxiliary consoles. Programs referencing `/dev/console` should be explicitly modified to use `syslog()` or `strlog()` if you want messages to be redirected to an auxiliary device.
- The `consadm` command runs a daemon to monitor auxiliary console devices. Any display device designated as an auxiliary console that disconnects, hangs up or loses carrier, is removed from the auxiliary console device list and is no longer active. Enabling one or more auxiliary consoles does not disable message display on the default console; messages continue to display on `/dev/console`.

Using Auxiliary Console Messaging During Run Level Transitions

Keep the following in mind when using auxiliary console messaging during run level transitions:

- Input cannot come from an auxiliary console if user input is expected for an `rc` script that is run when a system is booting. The input must come from the default console.
- The `sulogin` program, invoked by `init` to prompt for the superuser password when transitioning between run levels, has been modified to send the superuser password prompt to each auxiliary device in addition to the default console device.
- When the system is in single-user mode and one or more auxiliary consoles are enabled using the `consadm` command, a console login session runs on the first device to supply the correct superuser password to the `sulogin` prompt. When the correct password is received from a console device, `sulogin` disables input from all other console devices.
- A message is displayed on the default console and the other auxiliary consoles when one of the consoles assumes single-user privileges. This message indicates which device has become the console by accepting a correct superuser password. If there is a loss of carrier on the auxiliary console running the single-user shell, one of two actions might occur:
  - If the auxiliary console represents a system at run level 1, the system proceeds to the default run level.
  - If the auxiliary console represents a system at run level S, the system displays the `ENTER RUN LEVEL (0-6, s or S):` message on the device where the `init s` or `shutdown` command had been entered from the shell. If there isn't any carrier on that device either, you will have to reestablish carrier and enter the correct run level. The `init` or `shutdown` command will not re-display the run-level prompt.
- If you are logged in to a system using a serial port, and an `init` or `shutdown` command is issued to transition to another run level, the login session is lost whether this device is the auxiliary console or not. This situation is identical to releases without auxiliary console capabilities.

- Once a device is selected as an auxiliary console using the `consadm` command, it remains the auxiliary console until the system is rebooted or the auxiliary console is unselected. However, the `consadm` command includes an option to set a device as the auxiliary console across system reboots. (See the following procedure for step-by-step instructions.)

Using the `consadm` Command During an Interactive Login Session

If you want to run an interactive login session by logging in to a system using a terminal that is connected to a serial port, and then using the `consadm` command to see the console messages from the terminal, note the following behavior:

- If you use the terminal for an interactive login session while the auxiliary console is active, the console messages are sent to the `/dev/sysmsg` or `/dev/msglog` devices.
- While you issue commands on the terminal, input goes to your interactive session and not to the default console (`/dev/console`).
- If you run the `init` command to change run levels, the remote console software kills your interactive session and runs the `sulogin` program. At this point, input is accepted only from the terminal and is treated like it's coming from a console device. This allows you to enter your password to the `sulogin` program as described in "Using Auxiliary Console Messaging During Run Level Transitions" on page 323.

  Then, if you enter the correct password on the (auxiliary) terminal, the auxiliary console runs an interactive `sulogin` session, locks out the default console and any competing auxiliary console. This means the terminal essentially functions as the system console.
- From here you can change to run level 3 or go to another run level. If you change run levels, `sulogin` runs again on all console devices. If you exit or specify that the system should come up to run level 3, then all auxiliary consoles lose their ability to provide input. They revert to being display devices for console messages.

  As the system is coming up, you must provide information to `rc` scripts on the default console device. After the system comes back up, the `login` program runs on the serial ports and you can log back into another interactive session. If you've designated the device to be an auxiliary console, you will continue to get console messages on your terminal, but all input from the terminal goes to your interactive session.

▼ How to Enable an Auxiliary (Remote) Console

The `consadm` daemon does not start monitoring the port until after you add the auxiliary console with the `consadm` command. As a security feature, console messages are only redirected until carrier drops, or the auxiliary console device is unselected. This means carrier must be established on the port before you can successfully use the `consadm` command.

For more information about enabling an auxiliary console, see the `consadm(1m)` man page.

1 Log in to the system as the `root` user.

2. **Enable the auxiliary console.**

   ```
   # consadm -a devicename
   ```

3. **Verify that the current connection is the auxiliary console.**

   ```
   # consadm
   ```

Example 19–3 Enabling an Auxiliary (Remote) Console

```
# consadm -a /dev/term/a
# consadm
 /dev/term/a
```

How to Display a List of Auxiliary Consoles

1. **Log in to the system as the `root` user.**

2. **Select one of the following steps:**

   a. **Display the list of auxiliary consoles.**

      ```
      # consadm
      /dev/term/a
      ```

   b. **Display the list of persistent auxiliary consoles.**

      ```
      # consadm -p
      /dev/term/b
      ```

How to Enable an Auxiliary (Remote) Console Across System Reboots

1. **Log in to the system as the `root` user.**

2. **Enable the auxiliary console across system reboots.**

   ```
   # consadm -a -p devicename
   ```

   This adds the device to the list of persistent auxiliary consoles.

3. **Verify that the device has been added to the list of persistent auxiliary consoles.**

   ```
   # consadm
   ```

Example 19–4 Enabling an Auxiliary (Remote) Console Across System Reboots

```
# consadm -a -p /dev/term/a
# consadm
/dev/term/a
```

▼ How to Disable an Auxiliary (Remote) Console

1. **Log in to the system as the `root` user.**
2. **Select one of the following steps:**
   a. **Disable the auxiliary console.**

      `# consadm -d` *devicename*

      or

   b. **Disable the auxiliary console and remove it from the list of persistent auxiliary consoles.**

      `# consadm -p -d` *devicename*
3. **Verify that the auxiliary console has been disabled.**

   ```
   # consadm
   ```

Example 19–5 Disabling an Auxiliary (Remote) Console

```
# consadm -d /dev/term/a
# consadm
```

Troubleshooting File Access Problems

Users frequently experience problems, and call on a system administrator for help, because they cannot access a program, a file, or a directory that they could previously use.

Whenever you encounter such a problem, investigate one of three areas:

- The user's search path may have been changed, or the directories in the search path may not be in the proper order.
- The file or directory may not have the proper permissions or ownership.
- The configuration of a system accessed over the network may have changed.

This chapter briefly describes how to recognize problems in each of these three areas and suggests possible solutions.

Solving Problems With Search Paths (`Command not found`)

A message of `Command not found` indicates one of the following:

- The command is not available on the system.
- The command directory is not in the search path.

To fix a search path problem, you need to know the pathname of the directory where the command is stored.

If the wrong version of the command is found, a directory that has a command of the same name is in the search path. In this case, the proper directory may be later in the search path or may not be present at all.

You can display your current search path by using the `echo $PATH` command.

Use the `type` command to determine whether you are running the wrong version of the command. For example:

```
$ type acroread
acroread is /usr/bin/acroread
```

▼ How to Diagnose and Correct Search Path Problems

1 Display the current search path to verify that the directory for the command is not in your path or that it isn't misspelled.

```
$ echo $PATH
```

2 Check the following:

- Is the search path correct?
- Is the search path listed before other search paths where another version of the command is found?
- Is the command in one of the search paths?

If the path needs correction, go to step 3. Otherwise, go to step 4.

3 Add the path to the appropriate file, as shown in this table.

Shell	File	Syntax	Notes
bash and `ksh93`	`$HOME/.profile`	`$ PATH=`*`$HOME/bin:/sbin:/usr/local/bin ...`* `$ export PATH`	A colon separates path names.

4 **Activate the new path as follows:**

Shell	Path Location	Command to Activate The Path
bash and ksh93	.profile	$ **. ./.profile**
	.login	*hostname*$ **source .login**

5 **Verify the new path.**

$ **which** *command*

Example 19–6 Diagnosing and Correcting Search Path Problems

This example shows that the mytool executable is not in any of the directories in the search path using the type command.

```
$ mytool
 -bash: mytool: command not found
 $ type mytool
 -bash: type: mytool: not found
 $ echo $PATH
 /usr/bin:
 $ vi $HOME/.profile
 (Add appropriate command directory to the search path)
 $ . $HOME/.profile
 $ mytool
```

If you cannot find a command, look at the man page for its directory path.

Changing File and Group Ownerships

Frequently, file and directory ownerships change because someone edited the files as superuser. When you create home directories for new users, be sure to make the user the owner of the dot (.) file in the home directory. When users do not own " . " they cannot create files in their own home directory.

Access problems can also arise when the group ownership changes or when a group of which a user is a member is deleted from the /etc/group database.

For information about how to change the permissions or ownership of a file that you are having problems accessing, see Chapter 7, "Controlling Access to Files (Tasks)," in *Oracle Solaris Administration: Security Services*.

Solving File Access Problems

When users cannot access files or directories that they previously could access, the permissions or ownership of the files or directories probably has changed.

Recognizing Problems With Network Access

If users have problems using the `rcp` remote copy command to copy files over the network, the directories and files on the remote system may have restricted access by setting permissions. Another possible source of trouble is that the remote system and the local system are not configured to allow access.

See "Strategies for NFS Troubleshooting" in *Oracle Solaris Administration: Network Services* for information about problems with network access and problems with accessing systems through AutoFS.

◆ ◆ ◆ CHAPTER 20

Troubleshooting Miscellaneous System and Software Problems (Tasks)

This chapter describes miscellaneous system and software problems that might occur occasionally and are relatively easy to fix. The troubleshooting process usually includes solving problems that are not related to a specific software application or topic, such as unsuccessful reboots and full file systems.

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

- "What to Do If Rebooting Fails" on page 331
- "What to Do If a System Hang Occurs" on page 333
- "What to Do If a File System Fills Up" on page 333
- "What to Do If File ACLs Are Lost After Copy or Restore" on page 334

What to Do If Rebooting Fails

If the system does not reboot completely, or if the system reboots and then crashes again, there might be a software or hardware problem that is preventing the system from booting successfully.

Cause of System Not Booting	How to Fix the Problem
The system can't find `/platform/‘uname -m‘/kernel/unix`.	You may need to change the `boot-device` setting in the PROM on a SPARC based system. For information about changing the default boot device, see Chapter 6, "Modifying Boot Parameters on a SPARC Based System (Tasks)," in *Booting and Shutting Down Oracle Solaris on SPARC Platforms*.

Cause of System Not Booting	How to Fix the Problem
The GRUB boot archive has become corrupted or the GRUB menu is lost. Or, the SMF boot archive service has failed. An error message is displayed if you run the `svcs -x` command.	Create a second boot environment that is a backup of the primary boot environment. In the event the primary boot environment is not bootable, boot the backup boot environment. Alternatively, you can boot from the live CD or USB media. For more information about creating and managing BEs, see *Creating and Administering Oracle Solaris 11 Boot Environments*
There's an invalid entry in the `/etc/passwd` file.	For information about recovering from an invalid `passwd` file, see "What to Do If You Forgot the Root Password or Problem That Prevents System From Booting" on page 332.
There's a hardware problem with a disk or another device.	Check the hardware connections: ■ Make sure the equipment is plugged in. ■ Make sure all the switches are set properly. ■ Look at all the connectors and cables, including the Ethernet cables. ■ If all this fails, turn off the power to the system, wait 10 to 20 seconds, and then turn on the power again.

If none of the above suggestions solve the problem, contact your local service provider.

What to Do If You Forgot the Root Password or Problem That Prevents System From Booting

If you forget the `root` password or experience another problem that prevents the system from booting, do the following:

- Stop the system.
- Follow the directions in "How to Boot a System for Recovery Purposes" in *Booting and Shutting Down Oracle Solaris on SPARC Platforms*.
- If the `root` password is the problem, remove the root password from the `/etc/shadow` file.
- Reboot the system.
- Log in and set the `root` password.

What to Do If a System Hang Occurs

A system can freeze or hang rather than crash completely if some software process is stuck. Follow these steps to recover from a hung system.

1. Determine whether the system is running a window environment and follow these suggestions. If these suggestions do not solve the problem, go to step 2.
   - Make sure the pointer is in the window where you are typing the commands.
   - Press Control-q in case the user accidentally pressed Control-s, which freezes the screen. Control-s freezes only the window, not the entire screen. If a window is frozen, try using another window.
   - If possible, log in remotely from another system on the network. Use the `pgrep` command to look for the hung process. If it looks like the window system is hung, identify the process and kill it.
2. Press Control-\ to force quit the running program and (probably) write out a `core` file.
3. Press Control-c to interrupt the program that might be running.
4. Log in remotely and attempt to identify and kill the process that is hanging the system.
5. Log in remotely, become root and then reboot the system.
6. If the system still does not respond, force a crash dump and reboot. For information about forcing a crash dump and booting, see "Forcing a Crash Dump and Reboot of the System" in *Booting and Shutting Down Oracle Solaris on x86 Platforms*.
7. If the system still does not respond, turn the power off, wait a minute or so, then turn the power back on.
8. If you cannot get the system to respond at all, contact your local service provider for help.

What to Do If a File System Fills Up

When the root (`/`) file system or any other file system fills up, you will see the following message in the console window:

```
.... file system full
```

There are several reasons why a file system fills up. The following sections describe several scenarios for recovering from a full file system. For information about how to routinely clean out old and unused files to prevent file systems from becoming too full, see Chapter 13, Managing Disk Use (Tasks).

File System Fills Up Because a Large File or Directory Was Created

Reason Error Occurred	How to Fix the Problem
Someone accidentally copied a file or directory to the wrong location. This also happens when an application crashes and writes a large `core` file into the file system.	Log in as superuser and use the `ls -tl` command in the specific file system to identify which large file is newly created and remove it. For information about removing `core` files, see How to Find and Delete core Files.

A `TMPFS` File System Is Full Because the System Ran Out of Memory

Reason Error Occurred	How to Fix the Problem
This can occur if `TMPFS` is trying to write more than it is allowed or some current processes are using a lot of memory.	For information about recovering from `tmpfs`-related error messages, see the `tmpfs`(7FS) man page.

What to Do If File ACLs Are Lost After Copy or Restore

Reason Error Occurred	How to Fix the Problem
If files or directories with ACLs are copied or restored into the `/tmp` directory, the ACL attributes are lost. The `/tmp` directory is usually mounted as a temporary file system, which doesn't support UFS file system attributes such as ACLs.	Copy or restore files into the `/var/tmp` directory instead.

Index

A

acquit option, fmadm command, 150–151
adding
  groups, 63
  run control script (how to), 136
  SMF information, 113–114
  user initialization files, 47
  users, 61–62
addpg option, svccfg command, 113–114
addpropvalue option, svccfg command, 113–114
address space map, 166
admin layer, description, 109–110
administering
  accounts, 60–61
  groups, 63
  users, 61–62, 62
administering remote print queues, configuring CUPS, 278–279
administrative layers (SMF), *See* layers (SMF)
advanced server configuration, CUPS, 278
aging user passwords, 41
alert message priority (for syslogd), 321
aliases, user login names vs., 35
all milestone (SMF), description, 114
application threads, 182, 184
ASR, FMA and, 145
ASRU, definition, 143–144
at command, 250, 251, 254
  -l option (list), 253
  -m option (mail), 251, 252
  automatic scheduling of, 241
  controlling access to, 251, 254
at command, controlling access to *(Continued)*
  overview, 238
  denying access, 254
  error messages, 255
  overview, 238, 239, 250
at.deny file, 251, 254
  description, 238
at job files, 250, 253
  creating, 251, 252
  deleting, 253
  description, 239
  displaying, 253
  location of, 239
  submitting, 250
atjobs directory, 241
  description, 238
Auto Service Request, FMA and, 145
automatic system activity data collection, 211, 212
automatic system activity reporting, 211, 212
Automatic System Reconfiguration Unit, *See* ASRU
automatic system task execution
  repetitive tasks, 247, 248
  single tasks, 250, 251, 254
automating system task execution, 238
automounting, user home directories, 40
auxiliary (remote) console, 322

B

baud rate
  how to set on ttymon terminal, 295–296

baud rate *(Continued)*
how to set with the `eeprom` command, 295
`bin` group, 35
boot behavior, how to modify in GRUB menu, 92
booting
displaying messages generated during, 319
error logging (SMF) and, 114
milestone (SMF) and, 114
running `sadc` command when, 212
booting a system, run level S, 74–75
booting a system to run level 0, shutdown state, 78
bringing a system to a shutdown state, 78

C

C shell, user initialization files and, 56
`catman` utility
creating index files for man page searches, 24–25
`-M` option, 25
`-w` option, 25
`CDPATH` environment variable, 51
changing
account defaults, 60–61
`crontab` files, 242
date, 161
message of the day, 161
priority, 177, 179
timesharing processes, 178, 179
scheduling classes, 178
user passwords
by user, 37, 38
frequency of, 38
`Command not found` error message, 327
commands (SMF), list of, 110–111
`configCCR` command, manual registration and, 101–102
configuration repository (SMF), *See* repository
configuring a print server, CUPS, 278
configuring CUPS, remote print queue administration, 278–279
`consadm` command, 324–325
disabling an auxiliary console, 326
displaying list of auxiliary consoles (how to), 325
enabling an auxiliary console, 324–325
`consadm` command, enabling an auxiliary console *(Continued)*
across system reboots, 325
console
auxiliary
enabling across system reboots, 325
console terminal, how to set the baud rate on, 295–296
console terminal baud rate, setting with the `eeprom` command, 295
controlling
access to `at` command, 238, 251, 254
access to `crontab` command, 247, 248
overview, 238
processes, 169–170
controlling file and directory access, 55
copying a printer configuration
using CUPS, 286, 287
core dump configuration, displaying with `coreadm`, 312
core file name pattern, setting with `coreadm`, 311
core files
automatically deleting, 250
`core` files
examining with proc tools, 314
finding and deleting, 234
core files
managing with `coreadm`, 310
`coreadm` command, 310
displaying core dump configuration, 312
managing core files, 310
setting a core file name pattern, 313
CPU (central processing unit)
displaying information on
time usage, 165, 180
high-usage processes, 180
crash dump directory, recovering from a full, 306
crashes, 320
customer service and, 301, 316
displaying system information generated by, 306, 318
examining crash dumps, 305, 306
procedure following, 315
rebooting fails after, 331–332
saving crash dump information, 301

crashes *(Continued)*
saving other system information, 318
creating
at jobs, 252
`at` jobs, 251
`crontab` files, 242, 243
`cron.allow` file, 246, 247, 248
`cron` daemon, 239, 241
`cron.deny` file, 246, 247
defaults, 246
`crontab` command, 247
controlling access to, 246, 247, 248
denying access, 246, 247
limiting access to specific users, 246, 247, 248
overview, 238, 246, 247
`cron` daemon and, 241
`-e` option (edit), 242, 243
`-l` option (list), 244
`-r` option (remove), 245, 246
`/var/adm` maintenance and, 318
daily tasks, 238
error messages, 249
files used by, 241
overview, 238, 239
quitting without saving changes, 243
scheduling of, 241
`crontab` files
creating, 242, 243
creating and editing, 237–238
defaults, 240
deleting, 245, 246
denying access, 247
description, 241
displaying, 244
editing, 242, 243
location of, 240
removing, 245–246
syntax, 241, 242
`.cshrc` file, customizing, 56
CUPS
administering remote print queues, 278–279
configuring printer properties, 283–285
how to enable and disable a printer, 288
how to manage print jobs, 288–289
CUPS *(Continued)*
how to modify printer properties, 285–286
how to rename or copy a printer, 286, 287
how to share and unshare a printer, 287–288
selecting a print device, 280–281
CUPS GUI
deleting a printer, 287
setting up a local printer, 281–283
using to select a print device, 280–281
CUPS print server, configuring advanced settings, 278
customer service, sending crash information, 316
customizing
system message logging, 320
system message logging (how to), 322

D

`daemon` group, 35
daily tasks (scheduling with `crontab`), 238
`debug` log level, SMF, 110
default run level, definition, 115
defaults
message of the day, 161
`nice` number, 179
setting for users and roles, 60–61
defects (FMA)
displaying information about, 145–149
notification of, 145
repairing, 149–151
`degraded` SMF service state, description, 106
`delcust` option, `svccfg` command, 114
delegated restarters (SMF), 112
deleting
`at` jobs, 253
`core` files, 234
`crontab` files, 245, 246
finding and deleting old/inactive files, 231
log files, 243
old/inactive files, 239
SMF information, 114
temporary files, 233
users, 62
deleting a printer, using CUPS, 287
`delpropvalue` option, `svccfg` command, 114

dependencies (SMF), description, 105
determining
  run level (how to), 73
  system's run level (how to), 116
device, how to select when using CUPS, 280–281
df command, 192
  -h option, 225
  -k option (kilobytes), 192
  -t option (total blocks), 226
  examples, 192, 225
  overview, 192, 224
directories
  controlling access to, 55
  current working directory for processes, 166
  displaying information about, 227, 228, 230
  home, 38
  PATH environment variable and, 52, 53
  size of, 230
  skeleton, 47
  temporary, clearing out, 231, 233
disabled SMF service state, description, 106
disabling
  an auxiliary console with the consadm command, 326
  Oracle Configuration Manager, 101
  run control script (how to), 137
disabling a printer, using CUPS, 288
disk drives
  displaying information about
    free disk space, 192
  finding and deleting old/inactive files, 243
disk space
  displaying information about
    df command, 192
    directory sizes, 230
    file sizes, 227, 228, 230
    mount point, 193
  finding and deleting old/inactive files, 231, 235
  finding files exceeding a size limit, 229
  finding large files, 228, 229
dispadmin command, overview, 174
display
  date and time, 158
  host ID, 156
display *(Continued)*
  system's installed memory, 156–157
displaying
  at jobs, 253
  booting messages, 319
  core dump configuration with coreadm, 312
  crash information, 306, 318
  crontab files, 244
  directory information, 227, 228, 230
  file information
    file size, 227, 228
    listing newest, 231
    using the du command, 230
  FMA information, 145–149
  linked libraries, 166
  LWP information, 166
  priority information, 165, 175
  process information (how to), 168–169
  scheduling class information, 165, 174, 175
  size of files, 227–228
  system activity information, 195, 213
  system information
    commands for, 158
  user mask, 55
displaying a system's physical processor type, psrinfo -p, 158–159
displaying product name information, prtconf command, 156
dmesg command, 319
du command, 230
dumpadm, managing system crash information, 302

E

editing
  crontab files, 242, 243
eeprom command
  how to use to set boot parameters
    GRUB, 91–92
  using to set the baud rate on the ttymon terminal, 295
emCCR command, changing data collection, 102

enabling
an auxiliary console with consadm command, 324–325
auxiliary console across system reboots, 325
Oracle Configuration Manager, 101
enabling a printer, using CUPS, 288
encryption, 41
environment variables
LOGNAME, 52
PATH, 52
SHELL, 52
TZ, 52
errlog log file, 151
error logging (SMF), description, 110
error messages
at command, 255
crash messages, 319
crash related, 318
crontab command, 249
customizing logging of, 320
log file for, 315, 318
priorities for, 321
sources of, 320
specifying storage location for, 318, 320
/etc/cron.d/at.deny file, 251, 254
/etc/cron.d/cron.allow file, 246, 247, 248
/etc/cron.d/cron.deny file, 246, 247
/etc files
user account information and, 39
/etc/init.d directory, 136
/etc/inittab file
entry description, 117
example of default, 118
/etc/passwd file
description, 41
fields in, 41
user ID number assignment and, 35
/etc/shadow file, description, 41
/etc/svc/profile/site profiles, 107
/etc/syslog.conf file, 320
examining a core file, with proc tools, 314
executing routine tasks automatically (overview), 238
/export/home file system, 39

F

failed SMF boot archive service, troubleshooting GRUB based booting, 301
fast reboot
how to initiate, 81
how to initiate on x86 platforms, 82
Fault Management Architecture, *See* FMA
fault management resource identifier, *See* FMRI
faults (FMA)
displaying information about, 145–149
notification of, 145
repairing, 149–151
fcntl information, 166, 168
Field Replaceable Unit, *See* FRU
file or group ownership, solving file access problems, 328
file systems
disk space usage, 192
mount point, 193
files
checking access operations, 195, 196
controlling access to, 55
deleting
See deleting
displaying information about
listing, 227, 228
size, 227, 228, 230
displaying size of, 227–228
finding files exceeding a size limit, 229
for setting search path, 327
fstat and fcntl information display, 166, 168
size of, 227, 228, 230
find command
core files, 234
finding files exceeding a size limit, 229
old/inactive files, 231, 232
finding
and deleting old/inactive files
See deleting
files exceeding a size limit, 229
large files, 228, 229
fltlog log file, 151
FMA
displaying information, 145–149

FMA *(Continued)*
fault statistics, 151–152
log files, 151
notification, 145
overview, 143–144
repairing faults or defects, 149–151
`fmadm` command
example, 145–149
options, 150
overview, 149–151
`fmd` daemon, overview, 143–144
`fmdump` command
example, 147–148
FMA log files and, 151
FMRI, description, 105–106
`fmstat` command, example, 151–152
format of the man page sections, 29
FRU, definition, 143–144
`fsck` command, 239
`fstat` information, 166, 168
full-text search
man pages
`-K` *keywords* option, 25

G

`general` property group, description, 112
generating index files for man page searches,
`svc:/application/man-index:default` service, 23
GIDs, 35
assigning, 37
definition, 36
large, 36
global core file path, setting with `coreadm`, 310
global priorities
defined, 174
displaying, 175
`group` file
description, 41
fields in, 44
group ID numbers, 35, 36, 37
`groupadd` command, 47
adding group, 63
`groupdel` command, 47
`groupmod` command, 47
groups
adding, 63
changing primary, 37
default, 37
description, 36
description of names, 36
displaying groups a user belongs to, 37
guidelines for managing, 36, 37
ID numbers, 35, 36, 37
names
description, 36
naming services and, 37
primary, 36, 37
secondary, 36, 37
storage of information for, 41, 44
UNIX, 36
`groups` command, 37
GRUB-based booting, modifying the GRUB kernel
usage at boot time, 92
GRUB based booting
system crashes
failed SMF boot archive service, 301
GRUB menu entries, preserving Linux information, 93

H

home directories, removing, 62
`HOME` environment variable, 51
`/home` file system, user home directories and, 39

I

ID numbers
group, 35, 36, 37
user, 35, 36
index files for man page searches, using the `catman` utility, 24–25
`inetadm` command, description, 111
`infolog_hival` log file, 151
`infolog` log file, 151

init states
See run level
See run levels
initialization files, system, 40
initiating a fast reboot of the system
(how to), 82
how to, 81
`iostat` command
basic information display, 190
overview, 189

K

kernel thread
scheduling and, 165
structures, 165, 183
killing processes, 166, 170
klwp structure, 183
ksh93 shell, user initialization files and, 47
kthread structure, 183

L

`LANG` environment variable, 51, 54
large files, 229
layers (SMF), description, 109–110
`LC` environment variables, 54
`legacy_run` SMF service state, description, 106
`/lib/svc/manifest` files, overview, 106–107
library interfaces, SMF, 111
Linux menu entry, updating `menu.lst` file, 93
`listcust` option, `svccfg` command, 113
listing
files and directories, 227, 228, 231
processes, 167
processes being executed, 167
`listpg` option, `svccfg` command, 113
`listprop` option, `svccfg` command, 113
local printer
how to set up
CUPS, 281–283
local server configuration, CUPS, 278
`locale` environment variable, 51
log files, deleting automatically, 243
`logadm` command, FMA and, 151
`.login` file, customizing, 56
login names (user), description, 34
`LOGNAME` environment variable, 52
`ls` command
checking directory sizes, 227
`-l` option (size in bytes), 228
`-s` option (size in blocks), 228
`-t` option (newest files), 231
LWPs (lightweight processes)
defined, 182
displaying information about, 166
processes and, 182, 183
structures for, 183

M

mail aliases, user login names vs., 35
`MAIL` environment variable, 52
`maintenance` SMF service state, description, 106
`man` command
how to display a man page, 24
searching man pages, 23–24
man page searches
specifying the *SECTNAME*:*keywords* option, 28
using the `man -k` command, 27
man page sections, format, 29
man pages
creating index files to enable searching, 24–25
how to display, 24
what's new, 23–24
managing print jobs, using CUPS, 288–289
managing serial ports with SAF, task map, 293
managing system crash information, with `dumpadm`, 302
`manifest` layer, description, 109–110
manifests (SMF), description, 106–107
`MANPATH` environment variable, 52
maximums
finding files exceeding maximum size, 229
`nice` number, 179
secondary groups users can belong to, 36
user ID number, 35

maximums *(Continued)*
user login name length, 40
user password length, 37
`mdb` utility, 305, 306
memory
example of displaying information about, 157
process structures and, 183
shared
process virtual memory, 184
virtual
process, 184
`menu.lst` file, how to add Linux entry, 93
message of the day (MOTD) facility, 161
`messages` file, 315, 320
`messages.`*n* file, 318
milestone (SMF)
booting and, 114
description, 104
minimums
`nice` number, 179
user login name length, 40
user password length, 37
modify printer properties, using CUPS, 285–286
modifying, SMF information, 113–114
modifying kernel usage in the GRUB menu, 92
monthly tasks (scheduling with `crontab`), 239
MOTD (message of the day) facility, 161
`motd` file, 161
motd file, 161
mounting
user home directories
automounting, 40
user home directories (how to), 65
multiuser level, *See* run level 3
multiuser run level, description, 115

N

names
group
description, 36
user login
description, 34
naming services
groups and, 37
user accounts and, 39, 41
networks, recognizing access problems, 329
new features
SMF, 103
`svcadm enable system/sar:default`
command, 212
`newgrp` command, 37
`nice` command, 178, 179, 180
`nice` number, 165, 179
NIS
user accounts and, 39, 41
`noaccess` user/group, 35
`nobody` user/group, 35
`none` milestone (SMF), description, 114

O

`offline` SMF service state, description, 106
`online` SMF service state, description, 106
Oracle Configuration Manager
data collection, 102
disabling, 101
enabling, 101
manual registration, 101–102
overview, 99–100

P

panic messages, 318
`passwd` command, assigning user password, 61–62
`passwd` file, 41
fields in, 41
user ID number assignment and, 35
passwords, assigning to users, 61–62
passwords (user)
aging, 41
changing
frequency of, 38
by user, 37, 38
choosing, 38
description, 37

passwords (user) *(Continued)*
encryption, 41
precautions, 37, 38
setting, 37
PATH environment variable
description, 52, 53
per-process core file path, setting with coreadm, 310
perf file, 212
performance
activities that are tracked, 184
automatic collection of activity data, 211, 212
file access, 195, 196
manual collection of activity data, 195, 213
process management, 166, 179, 182
reports on, 195
system activity monitoring, 184, 195, 211
tools for monitoring, 184
permissions, 55
pfiles command, 166, 168
pflags command, 166
pkill command, 166, 170
pldd command, 166
pmap command, 166
power-down run level, description, 115
primary groups, 36, 37
print device
selecting
CUPS, 280–281
print jobs, using CUPS to manage, 288–289
printer
how to delete
using CUPS, 287
printer properties
using CUPS, 283–285
using CUPS to modify, 285–286
printer setup, using CUPS GUI, 281–283
printers
enabling and disabling
CUPS, 288
sharing and unsharing
CUPS, 287–288
priocntl command
overview, 174
-c option (scheduling class designation), 178
priocntl command *(Continued)*
-i option (ID type), 177, 178
-l option (scheduling class display), 174
-m option (max/min priority), 177
-p option (priority designation), 177
-s option (priority upper limit/change priority), 177, 178
priority (process)
changing, 177, 179
timesharing processes, 177, 178, 179
designating, 177
displaying information about, 165, 175
global
defined, 174
displaying, 175
overview, 174, 179
scheduling classes and, 177
user-mode priority, 174
/proc directory, 165
proc structure, 165, 183
proc tools, examining a core file, 314
process file system (PROCFS), 165
processes
address space map, 166
application threads and, 182, 184
controlling, 169–170
current working directory for, 166, 168
defined, 182
displaying information (how to), 168–169
displaying information about
priocntl command, 174
displaying information on, 164
listing processes, 167
listing processes being executed, 167
LWPs, 166
priocntl command, 174
ps command, 164, 167, 175
displaying information with proc tool commands, 166
displaying information with proc tools, 165
fstat and fcntl information for open files, 166, 168
killing, 166, 170
libraries linked into, 166

processes *(Continued)*
nice number of, 165, 178, 179, 180
priority, 179
changing, 177, 179
changing timesharing process priority, 177, 178, 179
designating, 177
displaying information about, 165, 175
global priorities, 174, 175
overview, 174, 179
scheduling classes and, 174, 177
user-mode priority, 174
proc tool commands, 165
restarting, 166
runaway, 180
scheduling classes, 173
changing, 178
changing priority of, 177, 179
designating, 177
displaying information about, 165, 174
displaying information on, 175
priority levels and, 174, 177
signal actions, 166
stack trace, 166
stopping temporarily, 166
structures for, 165, 183
terminology, 182, 184
tool commands, 166
tracing flags, 166
trees, 166, 168
troubleshooting, 180
PROCFS (process file system), 165
product name for a system, displaying with prtconf command, 156
.profile file, customizing, 56
profiles (SMF), description, 107
programs, disk-dependency of, 196
properties
configuring a printer
CUPS, 283–285
properties (SMF), description, 112
property groups (SMF), description, 112
prtconf command, 157
displaying a system's product name, 156
ps command, 164, 167
fields reported, 165
overview, 164
-c option (scheduling class), 165, 180
-ecl option (global priority), 175
-ef option (full information), 167
PS1 environment variable, 52
pseudo-ttys, 35
pseudo user logins, 35
psig command, 166
psrinfo command, examples, 148
psrinfo command option to identify chip multithreading features, psrinfo -p, 158
pstack command, 166
ptime command, 166
ptree command, 166, 168
pwait command, 166
pwdx command, 166, 168

Q

quiet log level, SMF, 110

R

real-time processes, changing class of, 178
reboot run level, description, 116
rebooting, fails after crash, 331–332
recognizing network access problems, 329
recovering from a full crash dump directory, 306
remote print queues, configuring CUPS, 278–279
removing, crontab files, 245–246
renaming a printer
using CUPS, 286, 287
repaired option, fmadm command, 150
repairing, FMA faults or defects, 149–151
repetitive system tasks, 247
replaced option, fmadm command, 150
repository (SMF)
backups of, 108
description, 104, 107
restarter property group, description, 112
restarters (SMF), 112

restarters (SMF) *(Continued)*
description, 103
restarting, processes, 166
`rm` command, 232, 233
`roleadd` command, 47
setting account defaults, 60–61
`roledel` command, 47
`rolemod` command, 47
run control scripts
adding (how to), 136
disabling (how to), 137
starting and stopping services, 135
run level
0 (power-down level), 72
1 (single-user level), 72
2 (multiuser level), 72
3 (multiuser with NFS), 73
6 (reboot level), 73
default run level, 72
definition, 72, 115
determining (how to), 73, 116
multiuser with NFS
what happens when system is brought to, 118
s or S (single-user level), 72
run level 0, shutdown state, 78
run level 3
multiuser with NFS
booting to, 73
runaway processes, 180

S

`sa1` command, 211
`sa2` command, 211, 212
`sadc` command, 211, 212, 213
`sa`*add* file, 212
`sar` command, 195, 213
description of all options, 213
options listed, 213
overview, 195, 213
`-a` option (file access), 195, 196
`-A` option (overall performance), 211, 213
`-b` option (buffers), 196
`-c` option (system calls), 198
`sar` command *(Continued)*
`-e` option (ending time), 213
`-f` option (file to extract data from), 213
`-i` option (interval), 213
`-m` option (interprocess communication), 203
`-p` option (page-in/page faults), 204
`-q` option (queue), 205, 206
`-r` option (unused memory), 206
`-s` option (starting time), 213
`-u` option (CPU usage), 207
`-v` option (system tables), 208
`-y` option (terminal devices), 210
saving crash dump information, 301
scheduling
See also `crontab` command, `at`command
one-time system tasks, 239, 250
repetitive system tasks, 238, 240
scheduling classes, 173
changing, 178
changing priority of, 177, 179
designating, 177
displaying information about, 165, 174, 175
priority levels and, 174, 177
search path, files for setting, 327
searching man page `NAME` subsection, using the `man -k` command, 27
searching man pages, using the `man` command, 23–24
secondary groups, 36, 37
SECTNAME:*keywords* option, of the `man -K` command, 28
security
`at` command, 251
`crontab` command, 247
user ID number reuse and, 36
service (SMF), description, 104
service configuration repository, *See* repository
Service Management Facility, *See* SMF
service states (SMF), description, 106
`setenv` option, `svccfg` command, 113–114
setting, a core file name pattern with `coreadm`, 313
Setting boot parameters by using `eeprom` command, GRUB based booting, 91–92
setting the baud rate on the `ttymon` console terminal, how to, 295–296

shadow file
description, 41
fields in, 44
shared memory, process virtual memory, 184
sharing a printer, using CUPS, 287–288
SHELL environment variable, 52
shells, user initialization files and, 56
shutdown command, shutting down a server (how to), 77
shutdown state, run level 0, 78
Simple Mail Transfer Protocol (SMTP), FMA and, 145
Simple Network Management Protocol (SNMP), FMA and, 145
single-user level, *See* run level s or S
single-user run level, description, 115
single-user state
booting a system
run level S, 74–75
site initialization files, 48
site-profile layer, description, 109–110
size
directory, 230
file, 227, 228, 230
skeleton directories (/etc/skel), 47
SMF
adding information, 113–114
booting and, 114
commands, 110–111
delegated restarters, 112
deleting information, 114
error logging, 110
library interfaces, 111
overview, 103
properties, 112
viewing information, 113
SMTP, FMA and, 145
snapshots (SMF), description, 109
SNMP, FMA and, 145
staff group, 37
start property group, description, 112
starting and stopping services, 135
stopping, processes temporarily, 166
stty command, 54
svc:/application/man-index:default service, generating index files for man page searches, 23
svc.startd daemon, description, 111–112
svcadm command, description, 111
svcadm enable system/sar:default command, 212
svccfg command
description, 111, 113
svcprop command
description, 111, 113
svcs command, description, 111
sys crontab, 212
syslog.conf file, 320
syslogd daemon, 318
system accounts, 35
system activities
automatic collection of data on, 211, 212
list of activities tracked, 184
manual collection of data on, 213
system crash information, managing with dumpadm, 302
system initialization files, 40
system message logging (customizing), 320
system messages
customizing logging (how to), 322
specifying storage location for, 318
system-profile layer, description, 109–110
system resources
monitoring, 251
automatic, 251
crashes, 320
overview, 182
system state
multiuser with NFS
booting to, 73
system tasks
See also crontab command, at command
scheduling
one-time tasks, 239, 250
repetitive tasks, 238, 240
scheduling automatically, 238

T

technical support
  crash dump analysis, 301
  sending crash information, 316
temporary directories, 231, 233
TERM environment variable, 52
terminals, process controlling, 165
TERMINFO environment variable, 52
time
  CPU usage, 165, 180
  processes accumulating large amounts of CPU time, 180
time zone environment variable, 52
timesharing processes
  changing scheduling parameters, 177
  priority of
    changing, 177, 178, 179
    overview, 174
    range of, 174
tools
  for displaying process information, 165
  process, 166
  system performance monitoring, 184
tracing flags, 166
troubleshooting
  processes, 180
troubleshooting system crashes
  GRUB
    boot archive service fails on reboot, 301
ttys (pseudo), 35
ttytype pseudo user logins, 35
TZ environment variable, 52

U

UIDs
  assigning, 36
  definition, 35
  large, 36
umask command, 55
uninitialized SMF service state, description, 106
UNIX groups, 36
UNIX systems (crash information), 301
unsetenv option, svccfg command, 114
unsharing a printer, using CUPS, 287–288
user accounts, 34
  description, 34
  guidelines for, 40
  ID numbers, 35, 36
  login names, 34
  naming services and, 39, 41
  setting up
    information sheet, 59
  storage of information for, 39
user home directories
  customized initialization files in, 47
  description, 38
  mounting
    automounting, 40
  mounting (how to), 65
  nonlocal reference to ($HOME), 39, 48
user ID numbers, 35, 36
user initialization files
  customizing, 47, 56
    adding customized files, 47
    avoiding local system references, 48
    overview, 48
    shell variables, 52
    site initialization files, 48
    user mask setting, 55
  description, 39, 40
  shells and, 56
user login names, description, 34
user logins (pseudo), 35
user mask, 55
user-mode priority, 174
user processes
  changing priority, 178, 179
  priority of, 174
user structure, 183
useradd command, 46
  adding user, 61–62
  setting account defaults, 60–61
userdel command, 46
  deleting user, 62
usermod command, 46
users
  adding, 61–62, 62

users *(Continued)*
removing home directories, 62
setting account defaults, 60–61
`/usr/adm/messages` file, 315
`/usr/bin/mdb` utility, 305
`/usr/lib/fm/fmd/fmd` daemon, overview, 143–144
`/usr/proc/bin` directory, 165, 166
`/usr/sbin/configCCR` command, manual registration and, 101–102
`/usr/sbin/emCCR` command, changing data collection, 102
`/usr/sbin/fmadm` command, example, 145–149
`/usr/sbin/fmdump` command
example, 147–148
FMA log files and, 151
`/usr/sbin/fmstat` command, example, 151–152
`/usr/sbin/logadm` command, FMA and, 151
`/usr/sbin/psrinfo` command, examples, 148
`/usr/sbin/svccfg` command, description, 113
`uucp` group, 35

V

`/var/adm` directory, controlling size of, 232
`/var/adm/messages` file, 315, 320
`/var/adm/messages.`*n* file, 318
`/var/adm/sa/sa`*dd* file, 212
`/var/fm/fmd` log files, 151
`/var/spool/cron/atjobs` directory, 238, 239, 241
`/var/spool/cron/crontabs` directory, 240, 241
`/var/spool/cron/crontabs/root` file, 240
`/var/spool/cron/crontabs/sys` crontab, 212
`/var/svc/manifest` files, overview, 106–107
`verbose` log level, SMF, 110
viewing, SMF information, 113
`vmstat` command, overview, 186

W

Watchdog reset ! message, 318
weekly tasks (scheduling with `crontab`), 239
`who` command, 73, 116