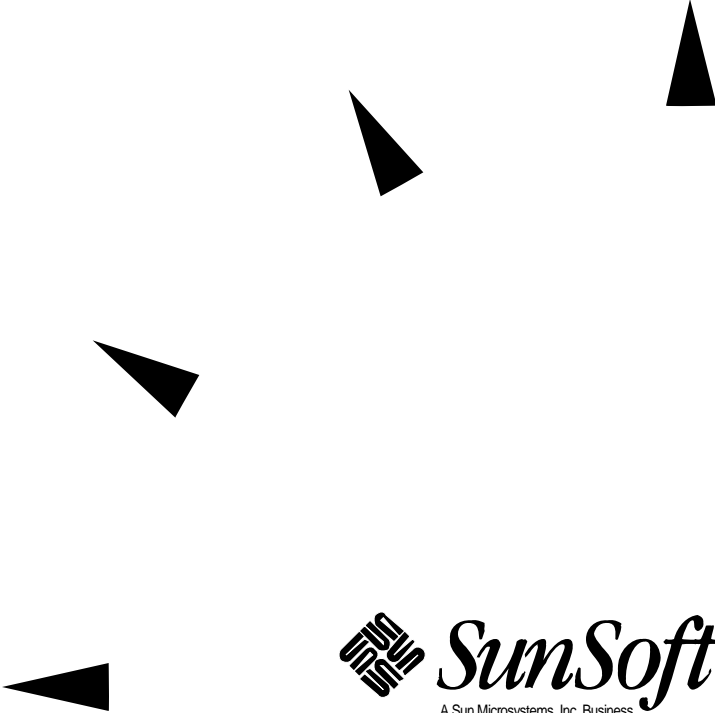


Solaris 1.x to Solaris 2.x Transition Guide

2550 Garcia Avenue
Mountain View, CA 94043
U.S.A.

[Part No: 802-1967-10](#)
[Revision A, November 1995](#)



© 1995 Sun Microsystems, Inc. 2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A.

All rights reserved. This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Portions of this product may be derived from the UNIX[®] system, licensed from UNIX System Laboratories, Inc., a wholly owned subsidiary of Novell, Inc., and from the Berkeley 4.3 BSD system, licensed from the University of California. Third-party software, including font technology in this product, is protected by copyright and licensed from Sun's suppliers.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and FAR 52.227-19. The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

TRADEMARKS

Sun, Sun Microsystems, the Sun logo, SunSoft, the SunSoft logo, Solaris, SunOS, OpenWindows, DeskSet, ONC, ONC+, and NFS are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd. OPEN LOOK is a registered trademark of Novell, Inc. PostScript and Display PostScript are trademarks of Adobe Systems, Inc.

All SPARC trademarks are trademarks or registered trademarks of SPARC International, Inc. in the United States and other countries. SPARCcenter, SPARCcluster, SPARCcompiler, SPARCdesign, SPARC811, SPARCengine, SPARCprinter, SPARCserver, SPARCstation, SPARCstorage, SPARCworks, microSPARC, microSPARC-II, and UltraSPARC are licensed exclusively to Sun Microsystems, Inc. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

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

X Window System is a trademark of X Consortium, Inc.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN. THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. SUN MICROSYSTEMS, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAMS(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.



Contents

Preface.....	xxi
<i>Part 1— Transition Information for Users and System Administrators</i>	
1. Introduction.....	3
Advantages of Migrating to the Solaris Operating Environment	3
Portability, Scalability, Interoperability, and Compatibility	5
Advantages for Large Organizations	6
Comparison of SVR4 and the Solaris Operating Environment	6
Additional Features in the Solaris Operating Environment	6
SVR4 Features Excluded From the Solaris Environment ..	9
2. Overview of Major Changes.....	11
Software Packages and Clusters.....	12
Package Administration	12
Disk Slices.....	13
Cylinder Groups	14
Device Naming.....	14

File Systems	15
Changes.	15
Pseudo File Systems	16
Added File Systems.	16
Kernel Configuration.	16
Kernel Layout.	17
Automounting	17
Admintool.	18
Network Information Service Plus (NIS+)	19
Print Subsystem	20
PrintTool	20
Command Changes.	20
Service Access Facility.	21
Volume Management.	22
3. Installation and Configuration Changes.	23
Installation Overview	23
What's New in Installation and Configuration	24
What to Do Before You Install Solaris Software.	25
Saving Disk Partition Information.	26
Making a List of System Components to Save.	27
Saving File System Information.	27
Making a List of Files and File Systems to Save.	28
Saving Metadevice Configuration Information	28
Deciding the Order of Installation for Networks.	29

Backing Up the File Systems	30
Saving SunOS Release 4.x System Data	30
Installing Solaris Software	31
Restoring Data	32
Restoring SunOS Release 4.x User Data	32
Restoring SunOS Release 4.x System Data	34
4. Using the Compatibility Packages	39
Why Port Applications?	40
SunOS/BSD Source Compatibility Package	40
Binary Compatibility Package	40
5. Security	43
Solaris 2.5 Security Features	44
/etc/passwd and /etc/shadow Files	44
/etc/default Files	45
Restricted Shells	45
Password Aging Changes	45
Automated Security Enhancement Tool (ASET)	46
Unbundled Security Options	47
Kerberos Security	47
SunShield	47
6. User Environment Administration	49
Selecting a Default Shell	49
Customizing User Environments	51
Window System	52

User and Group Administration	53
User and Group Administration Choices.....	53
Adding User Accounts.....	53
Using Mail	54
Using Document Tools	55
man Page Organization Differences.....	56
Customizing the man Command Search Path	57
whatis and windex Databases.....	58
Using the man Command	59
7. Device Administration.....	61
Device Naming Conventions	61
Convention for Disks.....	62
Convention for Tape Drives	63
Obtaining Disk Information	63
df Command	64
du Command	64
dtkinfo Command	64
devinfo Command	65
Adding Devices to the System	65
Using Volume Management.....	66
8. Startup and Shutdown.....	69
Booting	69
boot Command Changes.....	70
Booting From the PROM.....	70

Summary of Boot Differences.....	71
Using the <code>init</code> Command.....	72
<code>init</code> Command Changes.....	72
Changing System Run Levels.....	72
Shutting Down.....	74
Changes to the <code>shutdown</code> Command.....	75
Using the <code>fasthalt</code> and <code>fastboot</code> Commands.....	75
Using the <code>halt</code> and <code>reboot</code> Commands.....	76
9. File-System Administration	77
File-System Changes.....	78
Default File Systems and Directories.....	79
Virtual File-System Architecture	82
Supported File-System Types.....	82
Unsupported SVR4 File System Types	84
Generic File-System Commands	84
Directory and File Changes	87
<code>/dev</code> Directory.....	87
<code>/etc</code> Directory.....	87
<code>/sbin</code> Directory	92
<code>/usr</code> Directory.....	93
<code>/var</code> Directory.....	93
<code>/kernel</code> Directory	94
<code>/opt</code> Directory	94
<code>/sys</code> Directory.....	95

Using File System Administration Commands.....	95
Mounting File Systems and <code>autofs</code>	95
Monitoring File Systems.....	98
Sharing File Systems.....	99
Creating New File Systems.....	100
Checking File Systems.....	100
Backing Up and Restoring Files.....	101
10. Setting Up a Solaris 2.5 Server to Support SunOS Release 4.x Diskless/Dataless Clients	107
Adding SunOS Release 4.x Support to a Solaris 2.4 Server ...	107
Running <code>discover4x</code>	108
Setting Up the CD-ROM Drive for <code>install4x</code>	109
Running <code>install4x</code>	111
Running <code>convert4x</code>	115
11. Setting Up and Using Printers	117
Summary of Printing Differences	117
Print Commands and the Compatibility Package	118
Using Printer Commands.....	118
Using SunOS 5.5 Printer Administration Commands.....	119
Printing <code>troff</code>	119
12. Managing Terminals and Modems	121
Terminal and Modem Management	121
Service Access Facility (SAF)	122
What's a Port Monitor?.....	122

sacadm Command	123
pmdm Command.....	123
SAF Functions and Related Programs.....	124
13. Network Service Administration	125
Changes to TCP/IP	125
Changes to NFS	126
PPP.....	126
UUCP.....	126
Checkpoint Restart	128
User Job Grades	128
Limits File	128
Config File	129
Log Files	129
14. Using Name Services	131
Name Service Switch.....	132
NIS+.....	132
DNS	132
DNS and NIS+ Comparison	133
NIS and NIS+ Comparison.....	134
NIS Support in SunOS Release 5.5.....	135
Planning an NIS+ Upgrade.....	135
Installing NIS+ on a SunOS Release 4.x System.....	135
<i>Part 2—Transition Information for Developers</i>	
15. Compilers, Linkers, and Debuggers	139

Compilers	139
Compiler Option Differences	140
Linkers.	146
Link Editor Option Differences	146
Building Shared Libraries.	148
Building Executables.	148
Specifying Library Search Paths	149
Search Path Rules.	150
Version Numbering.	150
Examples.	151
Debuggers.	154
dbx and dbxtool	154
adb and kadb.	154
kadb Macros.	154
Debugging a Live Kernel	155
truss Command.	155
16. Tools and Resources	157
ioctl() Requests.	157
ptrace() Request Values	159
Libraries	161
Reorganized Libraries.	161
Shared Libraries.	161
Resource Limits	161
Using make	164

Using SCCS	165
Determining Application Compatibility	165
Well-Behaved Applications	165
Dynamically Linked and Statically Linked Executables	166
Packaging Applications	166
Packaging Utilities	166
Toolkits	167
OLIT	167
XView	167
Finding SunOS Release 4.x Tools	168
17. Networking and Internationalization	171
Networking	171
NIS, NIS+	172
nsswitch.conf File	172
NIT	172
Sockets	172
Internationalization	173
Character Support	173
Message Catalogs	174
Commands	174
Libraries	174
18. System and Device Configuration	177
System Configuration	177

Dynamically Loaded Kernel.....	177
Kernel Layout.....	178
config Command	178
/etc/system File.....	179
boot Command.....	179
Summary of Boot Differences.....	180
Device Configuration: The Reconfiguration Boot	181
Device Naming From a Developer's Perspective	181
/devices.....	181
/dev	182
Device Driver Naming	182
19. Device Drivers and STREAMS.....	185
Device Drivers and STREAMS Device Drivers	185
Device Driver Interfaces	186
devinfo Command	188
Porting Considerations	189
STREAMS	191
Solaris 2.x Driver Architecture.....	192
Device Driver Commands	193
A. Commands Reference Table	195
Using the Reference Table.....	195
Examples.....	196
The Commands Reference Table	197
B. System Calls Reference Table.....	237

Using the Reference Table.....	237
Examples.....	238
System Calls.....	239
C. Library Routines Reference Table.....	263
Using the Reference Table.....	263
Examples.....	264
Library Routines.....	265
D. System Files Reference Table.....	333
Using the Reference Table.....	333
System Files.....	334
E. / and /usr File Systems Changes.....	339
Layout of the / File System.....	339
Layout of the /usr File System.....	342
F. Quick Reference for Basic Changes.....	345

Figures

Figure 2-1	Disk Slices and Cylinder Groups	14
Figure 7-1	SunOS release 5.5 <code>prtvtoc</code> Command Output.....	65
Figure 9-1	Solaris Release 2.4 Default File Systems and Directory Hierarchy	80
Figure 9-2	SunOS Release 5.5 <code>/etc/vfstab</code> file	92

Tables

Table 2-1	SAF Functions and Associated Programs	22
Table 5-1	Files in /etc/default Directory	45
Table 6-1	Basic Features of the Bourne, C, and Korn Shells.	50
Table 6-2	Initialization Files for Bourne, C, and Korn Shells	51
Table 6-3	Default Home Directory Startup Files	51
Table 6-4	SunOS Release 5.5 man Page Directories	56
Table 6-5	New man Command Options	59
Table 7-1	SunOS Release 4.x and SunOS Release 5.5 Device Names . . .	63
Table 7-2	Location of CD-ROM and a Diskette With a File System	66
Table 7-3	Location of a CD-ROM and a Diskette Without a File System	67
Table 7-4	Volume Management Commands	67
Table 8-1	SunOS Release 5.5 Replacements for <code>reboot</code>	70
Table 8-2	Summary of Booting Differences	71
Table 8-3	SunOS Release 5.5 System Initialization Run Levels	73
Table 8-4	SunOS Release 5.5 Replacements for <code>shutdown</code>	74
Table 8-5	SunOS Release 5.5 <code>shutdown</code> Command Options.	75

Table 9-1	Default Solaris 2.4 File Systems and Directories	81
Table 9-2	Summary of File System Types	82
Table 9-3	SVR4 File System Types Not Supported	84
Table 9-4	Generic File System Administrative Commands	84
Table 9-5	Locations of File System Commands	86
Table 9-6	Additions to the /dev Directory	87
Table 9-7	Initialization Scripts and Their Run Control Files	88
Table 9-8	Additions to the /etc Directory	88
Table 9-9	/etc/vfstab File Field Names and Content	90
Table 9-10	Additions to the /usr Directory	93
Table 9-11	Files Changed in the /usr Directory	93
Table 9-12	Additions to the /kernel Directory	94
Table 9-13	SunOS Release 4.x and SunOS Release 5.5 mount Command Differences	96
Table 9-14	File and Directory Monitoring Commands	98
Table 9-15	ufsdump Command Options Not Available With the dump Command	102
Table 9-16	Tape Drive Density Characters in Tape Device Names	104
Table 9-17	cpio Options Added Since SunOS Release 4.x	104
Table 11-1	User Print Command Equivalents	118
Table 11-2	Printer Administration, Setup, and File Equivalents	119
Table 11-3	SunOS Release 5.5 troff commands	120
Table 12-1	SAF Functions and Associated Programs	124
Table 13-1	New SunOS Release 5.5 UUCP Files and Commands	127
Table 13-2	New SunOS Release 5.5 UUCP Log Files	129
Table 14-1	DNS and NIS+ Features and Benefits Compared	133

Table 14-2	NIS and NIS+ Features Compared	134
Table 15-1	Comparison of C Compiler Options	140
Table 15-2	Filename Extensions Used by the C Compiler	145
Table 15-3	Comparison of ld Options	146
Table 15-4	SunOS Release 4.x Linker Search Paths	150
Table 15-5	Solaris 2.5 Linker Search Paths With LD_LIBRARY_PATH= <i>dirlist1</i> 150	
Table 15-6	Solaris 2.5 Linker Search Paths With LD_LIBRARY_PATH= <i>dirlist1,dirlist2</i>	150
Table 15-7	Example Shared Libraries	151
Table 16-1	Unsupported ioctl() Requests	158
Table 16-2	Unsupported ttycom ioctl Requests	158
Table 16-3	ioctl() Support	158
Table 16-4	ptrace() Requests not Supported by Solaris 2.5 Software . .	160
Table 16-5	Resource Limits	162
Table 16-6	Comparison of Library Locations	162
Table 16-7	Commands for Creating Packages	167
Table 16-8	Commands for Adding and Removing Packages	167
Table 16-9	Commands for Providing Information About Packages	167
Table 16-10	Bundled Programming Tools	168
Table 16-11	New Programming Tools	169
Table 16-12	Unbundled Programming Tools	170
Table 17-1	xpg2lib Library Routine Locations	175
Table 18-1	Summary of Boot Differences	180
Table A-1	Example 1	196
Table A-2	Example 2	196

Table A-3	Example 3	197
Table A-4	Example 4	197
Table A-5	Commands Reference Table	198
Table B-1	System Calls Reference Table	239
Table C-1	Library Routines Reference Table	265
Table D-1	File Formats Reference Table	334
Table E-1	Directories in the / File System	339
Table E-2	Directories in the /usr File System	342
Table F-1	Basic Commands	345
Table F-2	Advanced Commands	346
Table F-3	Files and Directories	347
Table F-4	Daemons and Standard Processes	347

Preface

The Solaris™ 2.5 operating environment, SunSoft's™ distributed computing solution, is comprised of SunOS™ release 5.5 with ONC™, OpenWindows™, ToolTalk™, DeskSet™, and OPEN LOOK® as well as other utilities. *Solaris 1.x to Solaris 2.x Transition Guide* focuses on the differences between the SunOS release 4.x and SunOS release 5.5 operating systems for people already familiar with SunOS release 4.x. This guide also handles other aspects of the Solaris 2.5 operating environment that can help you through the transition.

If you are looking for more information about features now available with the Solaris 2.5 operating environment, see *Solaris 2.5 Introduction*.

The system administration tool covered in this book, Admintool, is part of the Solaris 2.5 product and can be used only for local system administration. System administration tools used to manage a network of systems are provided with the Solstice family of products.

Who Should Use This Guide

This guide can help users, system administrators, and software developers make the transition from a SunOS release 4.x computing environment to the Solaris 2.5 operating environment.

What to Expect From This Guide

The purpose of this guide is to give you an overview-level understanding of the differences between SunOS release 4.x and SunOS release 5.5 operating environments to make your transition to the Solaris 2.5 operating environment a smooth one. As a result, *Solaris 1.x to Solaris 2.x Transition Guide* covers a wide range of topics. Because it is not practical to list detailed procedures for tasks here, you will find references throughout this guide to publications in the Solaris 2.5 documentation set, where detailed information is available.

How This Guide Is Organized

This guide is divided into 2 parts with 19 chapters and 6 appendixes as outlined here.

Part 1: Transition Information for Users and System Administrators

You can use this part of the guide to help install Solaris 2.5 software, to understand changes to the local computing environment, and to understand changes to routine tasks.

This part of the guide contains the following chapters:

- **Chapter 1, “Introduction,”** discusses the benefits of migrating to the Solaris operating environment and summarizes the main differences between SVR4 and the Solaris operating environment.
- **Chapter 2, “Overview of Major Changes,”** is an overview of some of the principal changes between SunOS release 4.x and SunOS release 5.5. It provides background for topics in subsequent chapters, focusing on procedures, tools, and concepts that have changed between releases.
- **Chapter 3, “Installation and Configuration Changes,”** suggests what to consider to facilitate a smooth transition through software installation and post-installation so that SunOS release 4.x data can most easily be restored in the Solaris 2.5 operating environment.
- **Chapter 4, “Using the Compatibility Packages,”** discusses the SunOS/BSD Source Compatibility Package and the Binary Compatibility Package. These packages make the transition easier by enabling you to use SunOS release 4.x commands and applications during migration to the Solaris 2.5 operating environment.

-
- **Chapter 5, “Security,”** describes the major differences between SunOS release 4.x and Solaris 2.5 security, and points out how those changes might affect system administration procedures.
 - **Chapter 6, “User Environment Administration,”** describes differences in tasks used to set up a local user environment after installing the Solaris software. It includes discussions on setting up a default shell, customizing the user environment, the window system, and user and group administration. It also discusses changes regarding man pages.
 - **Chapter 7, “Device Administration,”** explains SunOS release 5.5 device naming conventions and discusses changes to device-related tasks such as getting information about disks, adding devices to a system, and using Volume Management.
 - **Chapter 8, “Startup and Shutdown,”** describes changes to procedures for booting and shutting down a system.
 - **Chapter 9, “File-System Administration,”** familiarizes you with changes to file system layout and the changes to file systems, virtual file systems, directories, and files. It also describes changes to file system administration.
 - **Chapter 10, “Setting Up a Solaris 2.5 Server to Support SunOS Release 4.x Diskless/Dataless Clients,”** discusses setting up servers for clients and it describes three programs, `discover4x`, `install4x`, and `convert4x`, that work together to help prepare a Solaris 2.5 server to serve SunOS release 4.x clients.
 - **Chapter 11, “Setting Up and Using Printers,”** describes how to set up and administer printers after you install Solaris 2.5 software. It also describes the changes to printer commands.
 - **Chapter 12, “Managing Terminals and Modems,”** describes terminal and modem management using Admintool and the Service Access Facility (SAF).
 - **Chapter 13, “Network Service Administration,”** outlines changes to the network facilities, TCP/IP and UUCP.
 - **Chapter 14, “Using Name Services,”** discusses NIS+ and the domain name system (DNS), and it compares NIS+ to NIS and DNS.

Part 2: Transition Information for Developers

This part of the guide concentrates on the changes that most affect developers. It describes these differences, points out similarities, and explains the implications for your programming environment.

This part contains the following chapters:

- **Chapter 15, “Compilers, Linkers, and Debuggers,”** discusses which capabilities have been added to or removed from compilers, linkers, and debuggers.
- **Chapter 16, “Tools and Resources,”** discusses changes to tools and resources for the development environment including changes to `ioctl()` requests, `ptrace()` request values, libraries, and the `make` and `SCCS` facilities. This chapter also describes how to determine application compatibility, how to use Solaris 2.5 packaging capabilities, and how to find SunOS release 4.x tools.
- **Chapter 17, “Networking and Internationalization,”** discusses Solaris 2.5 networking features as they relate to the programming environment and it discusses issues concerning the improved internationalization features.
- **Chapter 18, “System and Device Configuration,”** describes aspects of system and device configuration that have changed, including the dynamically loaded kernel and kernel layout, `config` and `boot` commands, and the `/etc/system` file.
- **Chapter 19, “Device Drivers and STREAMS,”** discusses device drivers issues such as changes to device driver interfaces, the `devinfo` command, porting considerations, `STREAMS`, and the Solaris 2.5 driver architecture.

Reference Appendixes

The following appendixes are comprised of reference tables showing SunOS 4.1 interfaces and their status in several operating systems. This information is useful to users, system administrators, and developers. The appendixes are:

- **Appendix A, “Commands Reference Table,”** compares SunOS release 4.x and SunOS release 5.5 commands.
- **Appendix B, “System Calls Reference Table,”** compares SunOS release 4.x and SunOS release 5.5 system calls.

- **Appendix C, “Library Routines Reference Table,”** compares SunOS release 4.x and SunOS release 5.5 library routines.
- **Appendix D, “System Files Reference Table,”** compares SunOS release 4.x and SunOS release 5.5 system files.
- **Appendix E, “/ and /usr File Systems Changes,”** compares SunOS release 4.x and SunOS release 5.5 system files.
- **Appendix F, “Quick Reference for Basic Changes,”** is a quick reference for changes in common commands, files and directories, and daemons and standard processes.

What Typographic Changes and Symbols Mean

Table P-1 describes the type changes and symbols used in this guide.

Table P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
AaBbCc123	The names of commands, files, and directories; on-screen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. system% You have mail.
AaBbCc123	What you type, contrasted with on-screen computer output	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> system% su Password: </div>
<i>AaBbCc123</i>	Command-line placeholder: replace with a real name or value	To delete a file, type <code>rm filename</code> .
<i>AaBbCc123</i>	Book titles, new words or terms, or words to be emphasized	Read Chapter 6 in <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> be root to do this.
Code samples are included in boxes and may display the following:		
%	UNIX C shell prompt	system%
\$	UNIX Bourne and Korn shell prompt	system\$
#	Superuser prompt, all shells	system#

man Page References

When commands, system files, or library routine names are first mentioned in the text, the number of the manual page section where the term is fully described is appended; for instance: `mv(1)`. The manual pages are in the *man Pages(1): User Commands*.

Related Books

For more information the Solaris 2.5 operating environment, see the following documentation:

- *Solaris 2.5 Introduction*
- *Solaris User's Guide*
- *Solaris Advanced User's Guide*
- *SPARC: Installing Solaris Software*
- *System Administration Guide, Volume I*
- *System Administration Guide, Volume II*
- *NIS+ and FNS Administration Guide*
- *NIS+ and DNS Setup and Configuration Guide*
- *NIS+ Transition Guide*
- *TCP/IP and Data Communications Administration Guide*
- *man Pages(1M): System Administration Commands* or on line in *SunOS Reference Manual*
- *Binary Compatibility Guide*
- *Source Compatibility Guide*
- *Writing Device Drivers*
- *Developer's Guide to Internationalization*
- *Driver Developer Kit Introduction*
- *Multithreaded Programming Guide*
- *Linker and Libraries Guide*
- *Programming Utilities Guide*

Part 1— Transition Information for Users and System Administrators

You can use this part of the guide to help install Solaris 2.5 software, and to understand changes to the local computing environment, and to understand changes to routine tasks.

Introduction



The Solaris operating environment enhances your system's capabilities with powerful tools and features. This introduction discusses the benefits of migrating to the Solaris operating environment and summarizes the principal differences between SVR4 and the Solaris operating environment.

This chapter contains the following sections:

<i>Advantages of Migrating to the Solaris Operating Environment</i>	<i>page 3</i>
<i>Comparison of SVR4 and the Solaris Operating Environment</i>	<i>page 6</i>

Advantages of Migrating to the Solaris Operating Environment

The UNIX standard, SVR4, accommodates the leading UNIX variants (System V, BSD, SunOS, and Xenix™), uniting the majority of the installed base of UNIX users. The Solaris operating environment, based on SVR4, gives software developers, system administrators, and end users the benefits of a standard operating system including broad compatibility, a growth path, and reduced time to market. It also delivers a functional and powerful product reflecting years of refinement. Among the many advantages the Solaris operating environment provides are portability, scalability, interoperability, and compatibility.

Although the foundation of the Solaris operating environment is based on SVR4, SunSoft has added extensive functionality in areas such as symmetric multiprocessing with multithreads, real-time functionality, increased security, and improved system administration.

The Solaris operating environment offers the following features:

- *SunOS release 5.5* is a full 32-bit operating system based on UNIX System V Release 4 (SVR4)
- *Cross-functional compatibility* enables SunOS release 5.5 to run on SPARC as well Intel 386, 486, Pentium and other DOS-compatible CPUs
- *Industry standards* including SVR4 and the ONC family of networking protocols
- *Graphical user interface* (GUI) in the OPEN LOOK® Window Manager
- *DeskSet* is a suite of easy-to-use personal and groupware productivity tools including Multimedia Mail, File Manager, Print Tool, Image Tool, Calendar Manager, and others
- *Calendar Manager* is a time management application that displays appointments and ToDo items at a glance and offers a multibrowse feature that makes it easy to schedule among a group
- *File Manager* provides a graphical and intuitive way to navigate to local and remote file systems
- *Image Tool* allows you to load, view and save images of over 40 different formats
- *OPEN LOOK Admintool* is the base for local system administration
- *Installation GUI* for easing install and update
- *Log-based filesystems* on servers
- *Advanced architecture* includes fully symmetric multiprocessing and sophisticated multithreading
- *Real-time priority scheduling* and a fully preemptible kernel, providing the benefits of open systems while meeting the requirements of control applications
- *Network Information Services Plus (NIS+)* is an upward-compatible version of the NIS name service with simpler hierarchical administration, improved security, and faster updates.
- *Standards conformance* for application developers interested in the benefits of application portability
- *Multimedia Mail* simplifies sending messages incorporating audio, graphics, and embedded files
- *Audio Tool* records and plays back spoken messages; they can even be attached to mail

Portability, Scalability, Interoperability, and Compatibility

The Solaris operating environment is portable, scalable, interoperable, and compatible.

Portability

SunSoft's SunOS is portable across multiple vendor platforms. Software conforming to an application binary interface (ABI) runs as shrink-wrapped software on all vendor systems with the same microprocessor architecture. This enables application developers to reduce software development costs and bring products to market quickly, and enables users to upgrade hardware while retaining their software applications and minimizing conversion costs.

Scalability

Over time, applications become more widely used, and require more powerful systems to support them. To operate in a growing environment, software must be able to run in a wide power range and must be able to take advantage of the additional processing power. SunSoft's operating system runs on machines of all sizes, from laptops to supercomputers.

Interoperability

Heterogenous computing environments are a reality today. Users purchase systems from many vendors to implement the solutions they need. Standardization and clear interfaces are critical to a heterogeneous environment, allowing users to develop strategies for communicating throughout their network. Solaris systems can interoperate with every popular system on the market today, and applications running on UNIX can communicate easily.

Compatibility

Computing technology continues to advance rapidly, but the need to remain competitive requires vendors to minimize their costs and to maximize their investments. SunSoft will ensure that as new technology is introduced, the existing software investment is preserved. Users can take advantage of today's solutions and still be compatible with tomorrow's technologies.

Advantages for Large Organizations

The Solaris operating environment provides a number of sound business reasons for transitioning to an industry-standard based UNIX operating system. Application development and maintenance costs are lower, and application portability is enhanced.

Comparison of SVR4 and the Solaris Operating Environment

This section describes the main differences between SVR4 and the Solaris operating environment. It points out features that the Solaris operating environment includes that are not available in SVR4 and a few SVR4 features that are not available in the Solaris operating environment.

Additional Features in the Solaris Operating Environment

The Solaris operating environment offers value-added components in addition to the SVR4-based operating system. These make computing easier and create new opportunities for users, system administrators, and developers.

In general, the merge of established UNIX variants into SVR4 and the Solaris operating environment was done by consolidating the existing functionality while maintaining compatibility for existing applications. As a result, features and commands were added to the product with few features being withdrawn.

Features for the User

For users, the Solaris operating environment incorporates a suite of powerful DeskSet applications to enhance personal productivity. All DeskSet applications rely on the drag-and-drop metaphor, enabling users to carry out complex UNIX commands with a mouse. Specifically, some of the features are:

- *A workspace manager.* Provides basic window management services (open, close, move, etc.), as well as tools which allow a user to tailor their workspace to their own personal needs.
- *Desktop integration services.* These include ToolTalk, drag and drop, and cut and paste, providing the foundation which allows applications to seamlessly integrate with one another.
- *Graphics libraries.* These include XGL, Xlib, PEX, and XIL, providing support for 2D and 3D graphics applications.

- *Calendar Manager.* A time management application that displays appointments and ToDo items for a day, week, or a month at a glance. It also contains a multibrowse feature that makes scheduling meetings among a group of users easy. Multiple calendars can be overlaid simultaneously to determine convenient meeting time slots at a glance.
- *Image Tool.* Allows you to load, view and save images of over 40 different formats including PICT, PostScript™, TIFF, GIF, JFIF, and many more.
- Other tools include a print tool, audio tool, shell tool, clock, and text editor.

Features for the System Administrator

For system administrators, the Solaris operating environment offers a variety of new tools to simplify the administration of a distributed computing environment. These include:

- *Device information.* Administrators can use these optional utilities to obtain information about installed devices including device names, attributes, and accessibility. Administration can be simplified by creating device allocation pools, a feature not previously found in UNIX systems.
- *File system administration.* These utilities enable administrators to create, copy, mount, debug, repair and unmount file systems, create and remove hard file links and named pipes, and manage volumes.
- *Interprocess communication.* Two interprocess communication utilities create, remove, and report on the status of the system's interprocess communication facilities (message queues, semaphores, and shared memory IDs). They provide information helpful in tuning the system.
- *Process management.* The process management utilities control system scheduling. Using these utilities, administrators can generate reports on performance, logins, disk access locations, and seek distances to better tune system performance. In addition, you can change the system run level, kill active processes, time the execution of commands, and change the default scheduling priorities of kernel, timesharing, and real-time processes.
- *System accounting.* The accounting utilities enable system administrators to track system usage by CPU, user, and process for better resource allocation.
- *System information.* These utilities report system memory and system configuration. The system administrator can use the utilities to change the names of the systems and the network node.

- *User and group management.* With these utilities, a system administrator can create and delete entries in group and password databases, specify default home directories and environments, maintain user and system logins, and assign group and user IDs. The utilities support both primary and supplementary user groups.
- *Admintool.* Admintool, which runs under the OpenWindows environment, provides system management facilities to help add hosts, manage the network, and many other routine tasks on local systems.
- *Auto configuration.* The Solaris operating environment has a dynamic kernel, which means that it loads drivers and other modules into memory when the devices are accessed. You no longer need to rebuild the kernel after installation, nor must you add or remove drivers.
- *Network Information Services Plus (NIS+).* An upward-compatible version of the NIS name service with simpler hierarchical administration, improved security, and faster updates.
- *Installation.* The Solaris operating environment has an install GUI to ease installation or upgrades. Automatic installations and upgrades are also possible over the network.
- *Security.* The automated security enhancement tool (ASET) is a utility that improves security by allowing system administrators to check system file settings including permissions, ownership, and file contents. ASET warns users about potential security problems and where appropriate, sets the system file permissions autonomically according the specified security level.

Features for the Developer

For application developers, the Solaris operating environment includes a variety of toolkits and features to simplify the development of complex applications with graphical user interfaces.

- *Multithreaded (MT) kernel.* MT provides for a symmetric multiprocessing kernel where multiple processors can execute the kernel at the same time. Applications can be structured as several independent computations rather than as one thread of control. Independent computations execute more efficiently because the operating system handles the interleaving of the independent operations. This benefit of multithreading is known as *application concurrency*.
- *STREAMS.* STREAMS is a framework for character input and output (I/O) and has been implemented throughout SVR4. It is a flexible framework that is easily customized for applications.

- *Expanded fundamental types.* ID data types (`uid`, `pid`, device IDs, etc.) and certain other data types are expanded to 32 bits. This improves the scalability of the operating system in large systems and for use in large organizations.
- *Device driver interfaces.* There are three types of interfaces for Solaris device drivers: Device Kernel Interface (DKI); Device Driver Interface/Device Kernel Interface (DDI/DKI), and Sun Device Driver Interface (Sun DDI). The DDI/DKI conformance means that device drivers have better source and binary compatibility across SPARC platforms so developers can write one driver to support a peripheral on all SPARC platforms.
- *Automatic device driver loading.* This makes drivers easier to install and devices easier to access.
- *Dynamic linking.* The Solaris application environment supports static and dynamic linking of libraries. The linker uses the version numbers of the libraries and executables to link applications with the proper libraries, routines, and interfaces.

SVR4 Features Excluded From the Solaris Environment

In a few instances, features in SVR4 were not include in the Solaris operating environment. These features are specific to AT&T[®] hardware, or features included primarily for backward compatibility with SVR3 features and therefore, are of little value to SunOS users.

The Solaris operating environment does not include the System V file system and associated utilities because of its limitations compared to the UNIX file system. The SVR4 boot file system was not included because of its additional burden of maintenance compared to the traditional SunOS model of booting.

The generic AT&T SVR4 model for device auto-configuration and for rebuilding kernels, was replaced with a fully dynamically configurable kernel better suited to the needs of present and future users of SPARC systems.

Because there is no installed base of SPARC XENIX[®] programs, the SPARC release of the Solaris operating environment does not include compatibility for XENIX applications.

The Solaris operating environment does not include the AT&T SVR4 `sysadm` utility. Because the `sysadm` menu utility was designed primarily for use with terminal devices on freestanding systems, SunSoft chose to concentrate its

efforts on tools with graphical user interfaces which simplify the administration of distributed systems across a network. The Solaris operating environment provides the utilities and configuration directories which underlie the SVR4 `sysadm` utility, but not the `sysadm` utility itself.

Overview of Major Changes



As you use the Solaris 2.5 operating environment, you will find similarities to the SunOS release 4.x operating environment; however, you will also notice some differences. The rest of this guide focuses on the procedures, tools, commands, and concepts that have changed between releases.

This chapter is an overview of some of the principal changes. It provides background for topics in subsequent chapters. Some topics receive sufficient coverage here, while others require more in-depth technical background. In the latter case, the section refers to a chapter that more fully describes the changes.

This chapter contains the following sections:

<i>Software Packages and Clusters</i>	<i>page 12</i>
<i>Disk Slices</i>	<i>page 13</i>
<i>Device Naming</i>	<i>page 14</i>
<i>File Systems</i>	<i>page 15</i>
<i>Kernel Configuration</i>	<i>page 16</i>
<i>Automounting</i>	<i>page 17</i>
<i>Admintool</i>	<i>page 18</i>
<i>Network Information Service Plus (NIS+)</i>	<i>page 19</i>
<i>Print Subsystem</i>	<i>page 20</i>
<i>Service Access Facility</i>	<i>page 21</i>
<i>Volume Management</i>	<i>page 22</i>

Software Packages and Clusters

Solaris 2.5 system software is delivered in units known as *packages*. A package is a collection of files and directories required for a software product. A *cluster* is a collection of packages.

The list below describes four clusters. Note that as you progress through the list, each cluster contains the software of the preceding cluster as well as additional software.

- *Core System Support* is the minimum software configuration; it contains only the software necessary to boot and run the Solaris 2.5 operating environment.
- *End User System Support* contains Core System Support plus end user support such as the OpenWindows windowing system and the related DeskSet application files; this cluster includes the recommended software for an end user.
- *Developer System Support* contains End User System Support plus the libraries, include files, and tools needed to develop software in the Solaris 2.5 operating environment. Compilers and debuggers are not included in the Solaris 2.5 operating environment.
- *Entire Distribution* contains the entire Solaris 2.5 environment.

For more information about this section's topics, see *System Administration Guide, Volume I*.

Package Administration

Software package management simplifies installing and updating software. Administration is simplified because the method for managing system software and third party applications is now consistent. The tools for creating software packages are in an application packaging tools library.

There are two tools you can use to install and remove packages:

- A graphical user interface program (see the `admintool(1m)` man page)
- The command-line utilities (see the `pkgadd(1M)` and `pkgrm(1M)` man pages)

Graphical User Interface (`admintool`)

You can install software on your local system or on a remote system with Admintool (started with the `admintool` command). The default location for the installation is the local system.

Use Admintool to:

- Look at the software installed on the local system
- Install or remove software on a local system

If you want to install or remove the software, you must run Admintool as superuser or as a user in the `sysadmin` group (group 14). You do not need to be superuser to look at the software packages that are already installed on a system.

Command-Line Utilities

You can use command-line utilities to install, remove, and check the installation of software packages. The commands are:

- `pkgadd(1M)` for installing a package
- `pkgrm(1M)` for removing a package
- `pkgchk(1M)` for checking the installation of a package
- `pkginfo(1M)` for listing the packages installed on a system

Disk Slices

A disk *slice* is composed of a single range of contiguous blocks, and is a physical subset of the disk (except for slice 2, which customarily represents the entire disk). Before you can create a file system on a disk, you must format and divide it into slices. See *System Administration Guide, Volume I* for complete information on installing and formatting disks, and dividing disks into slices.

A slice can be used as a raw device for swap space or to hold one and only one UFS file system. A disk can be divided into as many as eight slices. See *System Administration Guide, Volume I* for a list of customary disk slice assignments.

Cylinder Groups

You create a UFS file system on a disk slice, which is divided into one or more areas called *cylinder groups*. A cylinder group is composed of one or more consecutive disk cylinders (the set of tracks on a group of platters that have the same radial distance from the center of the platter). See *System Administration Guide, Volume I* for a complete description of disk geometry.

A *cylinder group map* is created for each cylinder group. The cylinder group map records the block usage and available blocks.

Figure 2-1 shows the relationship between disk slices and cylinder groups.

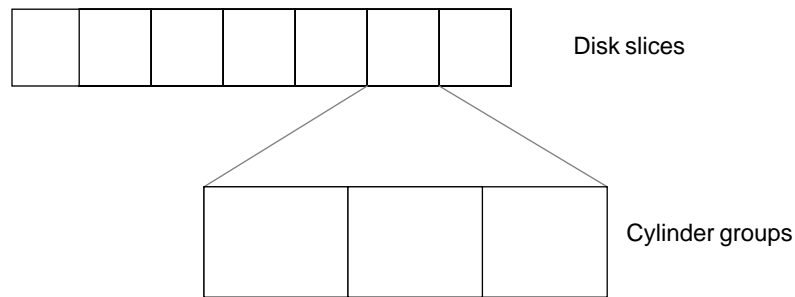


Figure 2-1 Disk Slices and Cylinder Groups

Device Naming

SunOS release 5.5 device names make it easier to infer certain device characteristics from a device name. SunOS release 4.x systems convey type, rather than device attributes, which makes it difficult for programs and scripts to derive necessary information about devices. SunOS release 5.5 conventions are slightly different from AT&T SVR4 device names because SunOS release 5.5 allows only eight partitions on a disk.

In addition, the `/dev` directory that contains the special device names has been changed from a flat directory to a hierarchical one, with a separate subdirectory for each category of device. For example, the location of disk device files is `/dev/dsk`, while raw disks are located in `/dev/rdsk` (see “Directory and File Changes” on page 87). For discussions on device-naming conventions, see “Device Naming Conventions” on page 61 and “Device Naming From a Developer’s Perspective” on page 181.

File Systems

SunOS release 5.5 and SunOS release 4.x file systems are similar, but there are changes in the locations and names of system directories and files. There are also new file systems, new pseudo file systems, and one directory is not used.

Chapter 9, “File-System Administration,” describes file system changes. *System Administration Guide, Volume I* describes file system concepts and administration in detail.

Changes

Some of the changes to file system locations and names are:

- The `/dev` directory has changed from a flat directory to a hierarchical one.
- The `/etc` directory contains system configuration information. Several files and subdirectories have been added, removed, or changed for SunOS release 5.5.
- The `/etc/vfstab` tab file replaces `/etc/fstab`.
- The `/etc/lp` directory replaces `/etc/printcap`.
- The SunOS release 5.5 `/sbin` directory contains the `rc` scripts used to alter system run levels as well as the `rds` script used to initialize the system prior to mounting file systems.
- The SunOS release 5.5 `/usr` directory contains sharable files and executables provided by the system.
- The `/var` directory contains files that change sizes during normal operation. Several files and subdirectories in the `/var` directory have been added, removed, or changed for SunOS release 5.5.
- The `/var/mail` directory replaces `/var/spool/mail`.
- The `/sys` directory is no longer needed because the kernel is dynamically loaded.
- The `terminfo` database replaces `termcap`.
- The `/RFS` file system has been removed.
- The kernel is called `unix` in SunOS release 5.5, and the kernel modules are stored in the `/kernel` directory.

Pseudo File Systems

Pseudo file system types are logical groupings of files that reside in disk-based systems. The TFS pseudo file system is not included in SunOS release 5.5.

The pseudo file systems in SunOS release 5.5 are:

- **CACHEFS** pseudo file system – can be used to improve performance of slow devices such as CD-ROM.
- **PROCFS** pseudo file system – resides in memory and contains a list of active processes, by process number, in the `/proc` directory. See the `proc(4)` manual page.
- **FDFS** pseudo file system – provides explicit names for opening files using file descriptors.
- **FIFOFS** pseudo file system – contains pipe files that give processes common access to data.
- **NAMEFS** pseudo file system – is used mostly by STREAMS for dynamic mounts of file descriptors on top of files.
- **SWAPFS** pseudo file system – is the default swap device when the system boots or you create additional swap space.

Added File Systems

The following file systems are included in the SunOS release 5.5 directory structure:

- The optional `/opt` file system – can be used to store third-party or unbundled software. If `/opt` is not a separate file system, it may be a symbolic link to `/usr/opt`.
- The `/vol` file system – provides the default file system for the Volume Management daemon, `vold(1M)`. See the `vols(7)` manual page.

Kernel Configuration

Unlike SunOS release 4.x, the SunOS release 5.5 kernel is dynamically configured. This means that you no longer need to rebuild it manually when you make changes to the system configuration. The kernel consists of a small static core and many dynamically loadable kernel modules. Drivers, file

systems, STREAMS modules, and other modules are loaded automatically as needed, either at boot time or at run time. These modules are unloaded when they are no longer in use. The `modinfo(1M)` command provides information about the modules currently loaded on a system.

The `modload(1M)` and `modunload(1M)` commands are still available in this release, but they perform differently. These commands have more limited usage, and are no longer sufficient to correctly install a loadable driver onto the system. `modunload` is similar to the SunOS release 4.x command, but it includes the capability to unload all unloadable (and not busy) modules as the following example illustrates.

```
# modunload -i 0
```

Chapter 18, “System and Device Configuration,” discusses these topics in more detail.

Kernel Layout

The contents of the kernel, which were formerly in a single file, `/vmunix`, are now contained in modules in a directory hierarchy. By default, the directory hierarchy is `/kernel` and `/usr/kernel`.

The directory search path for modules can be set by the `moddir` variable in the `/etc/system` file. See the `system(4)` man page in the *man Pages(1M): System Administration Commands*. Typically, `/kernel/unix` is the first portion of the kernel to be loaded. See the `kernel(1M)` manual page in the *man Pages(1M): System Administration Commands*.

The `/kernel/drv` directory is used for bootable devices and system-dependent software. The `/usr/kernel/drv` directory is used for all other drivers.

Automounting

SunOS 5.x supports automatic mounting for file systems shared through NFS. `autofs` runs in the background, mounting and unmounting remote directories on an as-needed basis, without user input. `autofs` starts automatically when a

system enters run level 3, or you can start it from a shell command line. See *NFS Administration Guide* for complete information on how to set up and administer `autofs`.

The following example shows that the `/home/bugs` directory was not mounted until it was accessed.

```
% mount | grep bugs
% cd /home/bugs
% mount | grep bugs
/tmp_mnt/home/bugs on bugs:/export/home/bugs intr/remote on Fri Mar 4 16:16:00 1994
```

Whenever a user tries to access a remote file or directory (using the `cd(1)` command for example), `autofs` mounts the file system to which that file or directory belongs. This remote file system remains mounted for as long as it is needed. If the remote file system is not accessed for a certain period of time, it is automatically unmounted. No mounting is done at boot time, and the user does not need to know the superuser password to mount a directory; users need not use the `mount` and `umount` commands, although they are still available.

`autofs` works with file systems specified in NIS maps or NIS+ tables. These maps or tables can be maintained as NIS, NIS+, or local files.

`autofs` maps or tables can specify several remote locations for a particular file. This way, if one of the servers is down, the automounter can try to mount from another system. You can specify which servers are preferred for each resource in the maps by assigning each server a weighting factor.

By default, the SunOS system software automounts `/home`.

See “Mounting File Systems and `autofs`” on page 95 for more detailed information.

Admintool

One of the major changes affecting system administration between SunOS release 4.x and SunOS release 5.x is the availability of `Admintool` with `OpenWindows`. This tool employs a graphical user interface to simplify tasks, such as managing users, hosts, printers, and serial devices. These tasks can be managed on local systems if the right access has been set up.

Admintool applications enable you to manage the following:

- Local system database files such as `aliases` and `netmasks`
- User account information, including tasks such as adding users, modifying password aging features, and removing user account information
- Local and remote printer setup
- Terminal and modem setup

Using a graphical user interface (GUI) like Admintool to perform the administration tasks has the following benefits:

- It is faster than using numerous SunOS commands to perform the same tasks
- System files are updated automatically without the risk of making editing errors in important system files
- The application programs interact with appropriate system daemons and notify you when the two are out of sync

Note – You do not need to be root to start Admintool, but you do need to be a member in the `sysadmin` group (use the `groups(1)` command to display your groups). Running `admintool` as root will restrict your changes to the local system.

To display Admintool, type the following command in any window.

```
$ admintool &
```

Network Information Service Plus (NIS+)

Note – SunOS release 5.x does not support NIS. Clients that use the NIS or DNS naming service must either use the NIS binary compatibility package or continue to use the SunOS release 4.x NIS or DNS server.

NIS+ is the network information service for Solaris networks. NIS data can be shared with NIS+ and the services can be combined to some extent. All commands and functions that use NIS are prefixed by the letters *yp*, as in `ypmatch(1)` and `ypcat(1)`. Commands and functions that use NIS+ are prefixed by the letters *nis*, as in `nismatch(1)` and `nischown(1)`.

NIS+ is a name service built on top of the ONC transport-independent remote procedure call (TI-RPC) interface. NIS+ has significant benefits compared to NIS in the areas of security, performance, scalability, and administration. Some of the advantages of using NIS+ are:

- NIS+ shares data with the NIS environments, allowing a smooth migration.
- Domains are hierarchical; you can create subdomains.
- You can use the name service switch (`/etc/nsswitch.conf`) to set which name service a system will try to use first – NIS+, NIS, or DNS.
- You can use the Database Manager to make changes to NIS+ tables for adding, modifying, deleting, and searching for information.

See Chapter 14, “Using Name Services,” in this guide and *NIS+ Transition Guide*, *NIS+ and FNS Administration Guide*, and *NIS+ and DNS Setup and Configuration Guide* for more information.

Print Subsystem

The print management commands have changed between SunOS release 4.x and the Solaris 2.5 operating environment. In the Solaris 2.5 operating environment, you can use command-line procedures or Admintool to set up printers, and you can use administrative commands or the PrintTool to control print jobs.

See Chapter 11, “Setting Up and Using Printers,” and *System Administration Guide, Volume II* for more information.

Users can accomplish the same basic tasks using the PrintTool or commands in a shell.

PrintTool

PrintTool is a software tool available through OpenWindows in the Solaris 2.5 user environment. It provides a graphical user interface through which a user can monitor printers and monitor and cancel print jobs.

Command Changes

The following list summarizes command changes:

- `lp(1)` replaces `lpr` and the `/etc/printcap` file

- `lpstat(1)` replaces `lpq`
- `cancel(1)` replaces `lprm`
- `troff(1)` requires a printer name
- `TEX`, `pscat` (C/A/T), and raster image filters are not available in Solaris 2.5

The `lp` service consists of several daemons, or processes, that monitor system work, a hierarchy of configuration files in the `/etc/lp` directory; and as a set of administrative commands.

Service Access Facility

The Service Access Facility (SAF) is the tool used for administering terminals, modems, and other network devices. In particular, the SAF enables you to:

- Add and administer `ttymon` and listen port monitors (using the `sacadm` command)
- Add and administer `ttymon` port monitor services (using the `pmadm` and `ttyadm` commands)
- Add and administer listen port monitor services (using the `pmadm` and `nlsadmin` commands)
- Administer and troubleshoot TTY devices
- Administer and troubleshoot incoming network requests for printing service
- Administer and troubleshoot the Service Access Controller (using the `sacadm` command)

The SAF is an open systems solution that controls access to system and network resources through TTY devices and local-area networks (LANs). The SAF offers well-defined interfaces that make it easy to add new features and configure existing ones.

The SAF is not a program. It is a hierarchy of background processes and administrative commands. The top-level SAF program is the SAC. The SAC controls port monitors that you administer through the `sacadm` command. Each port monitor can manage one or more ports.

You administer the services associated with ports through the `pmadm` command. While services provided through SAC may differ from network to network, SAC and the administrative programs `sacadm` and `pmadm` are not tailored to network types.

Table 2-1 illustrates the SAF control hierarchy. The `sacadm` command is used to administer the SAC, which controls the `ttymon` and `listen` port monitors.

Table 2-1 SAF Functions and Associated Programs

Function	Program	Description
Overall Administration	<code>sacadm</code>	Command for adding and removing port monitors
Service Access Controller	<code>sac</code>	SAF's master program
Port Monitors	<code>ttymon</code> <code>listen</code>	Monitors serial port login requests Monitors requests for network services
Port Monitor Service Administrator	<code>pmadm</code>	Controls port monitors' services
Services	<code>logins</code> ; remote procedure calls; etc	Services to which SAF provides access

The services of `ttymon` and `listen` are in turn controlled by `pmadm`. One instance of `ttymon` can service multiple ports and one instance of `listen` can provide multiple services on a network interface.

See Chapter 12, "Managing Terminals and Modems," for more information.

Volume Management

Beginning with the Solaris 2.2 software, a new layer of software manages CD-ROM and diskette devices — Volume Management. This software automates the interaction between you and your CD-ROMs and diskettes.

OpenWindows users will notice a slight change to the File Manager — it has been modified to use Volume Management to provide immediate user access to CD-ROMs and diskettes with file systems on them. See the *Solaris User's Guide* for more information on File Manager's new features.

There are also several new commands to help you administer Volume Management on your system.

For more information, see "Using Volume Management," in Chapter 7, "Device Administration".

Installation and Configuration Changes



Migrating from SunOS release 4.x to the Solaris 2.5 operating environment is a three-phase process that includes pre-installation (planning and backing up data), installation (installing Solaris), and post-installation (restoring data).

This chapter offers information about the pre-installation and post-installation phases for a single system or an entire network. It suggests what to consider to help facilitate a smooth transition through the migration to the Solaris operating environment. (See Chapter 10, “Setting Up a Solaris 2.5 Server to Support SunOS Release 4.x Diskless/Dataless Clients,” for information about creating an environment that serves both Solaris 2.5 and SunOS release 4.x clients.)

This chapter contains the following sections:

<i>Installation Overview</i>	<i>page 23</i>
<i>What to Do Before You Install Solaris Software</i>	<i>page 25</i>
<i>Installing Solaris Software</i>	<i>page 31</i>
<i>Restoring Data</i>	<i>page 32</i>

Installation Overview

Issues for the three phases of software migration are the same, whether you migrate the entire environment at once or just a small portion of it. There are differences in the tasks you will perform based on the type of system on which you are installing software. Those differences are detailed in *SPARC: Installing*

Solaris Software, along with steps to perform the actual installation. Read this chapter before referring to *SPARC: Installing Solaris Software* so you understand the migration-related issues before you start.

Here is an outline of the three-phase process for a migration to the Solaris computing environment:

- What to do before you install Solaris software
 - Save configuration information; know the current layout of the disk and file systems.
 - Save the file system information such as the mapping between file system names and device names.
 - Decide which files to save, including applications, unbundled software products, and administrative files that SunOS release 4.x needs to re-create its environment and recognize its files and applications.
 - Perform backups.
- Installing Solaris software

Install SunOS release 5.5 using installation instructions from *SPARC: Installing Solaris Software*.
- What to do after you install Solaris software
 - If the system is a server, set up a NIS+ master name server and prepare the server for SunOS release 4.x clients (see Chapter 10, “Setting Up a Solaris 2.5 Server to Support SunOS Release 4.x Diskless/Dataless Clients”).
 - Restore SunOS release 4.x user and system data.
 - If the system will serve both Solaris clients and SunOS release 4.x clients, you will need to run programs to prepare the server (see Chapter 10, “Setting Up a Solaris 2.5 Server to Support SunOS Release 4.x Diskless/Dataless Clients”).

Note that this chapter covers the installation process for different types of systems: servers, standalone systems, and dataless clients.

What’s New in Installation and Configuration

SunOS release 5.5 installation introduces a number of changes in the way administrators install software on existing systems and add new systems to the network.

- SunOS release 5.5 is distributed on compact disc (CD) only. This means you must have access to a CD-ROM drive before you can install the software. However, you can set up a system that has a CD-ROM drive to act as an install server for systems without local CD-ROM drives on the network.
- SunOS release 5.5 is bundled into modules called packages. You can select packages relevant to your system and control the amount of space each installation requires or specify the applications to be available on an individual system.

SunOS release 5.5 packages are grouped into clusters. This means that you can select a set of related packages without having to select each package separately.

- SunOS release 5.5 installation also provides a set software groups, which are groups of packages and clusters for typical users. You can select a software group to get systems running without selecting individual packages and clusters. This can be useful when you are first installing SunOS release 5.5 in a limited environment for testing. You can add or remove packages later as you gain more experience with the system.
- SunOS release 5.5 includes architecture-specific kernels, rather than the generic kernel configuration provided in earlier SunOS software releases. You will find the installed kernel in `/kernel/unix` instead of `/vmunix`.

What to Do Before You Install Solaris Software

Converting a SunOS release 4.x system to a Solaris 2.5 system involves more than just running the installation programs and loading the software. Usually, there is data on the SunOS release 4.x system that needs to be transferred to a Solaris 2.4 system. This data may be full file systems, such as `/home`, or locally customized system files, such as `/etc/hosts` or `/etc/passwd`.

No matter how you plan to handle the data transfer, you should back up all disk partitions by doing full dumps before you begin the installation process, just in case something should happen to any of the data. You should be cautious when installing Solaris 2.5 software. Because the device naming conventions are different in the Solaris 2.5 operating environment, it is possible to become confused and inadvertently choose the wrong disk when you install the Solaris 2.5 software. Backing up the file systems before you begin the installation procedure offers some protection should this sort of accident occur.

Note – Before you do anything else, back up all disk partitions.

Note about file system formats:

- If the Solaris 2.4 Extended Fundamental Types (EFT) are not used, the file system format used in SunOS release 4.x is upwardly compatible with and, in some cases, identical to that used in Solaris 2.4 software.
- If you are running SunOS 4.1.1 with QuickCheck or Backup Copilot installed or SunOS 4.1.2, the file system formats are identical.
- If you are running SunOS 4.1.1 without QuickCheck or Backup Copilot, SunOS 4.0.x or SunOS 4.1, the file systems are upwardly and backwardly compatible, although not identical in all cases.

Saving Disk Partition Information

Before you begin the installation process, you should save a hard copy (that is, on paper) of the system's existing disk partitions. If this information is saved online, it may be overwritten during installation. Saving the existing disk partition information serves as a reference for many decisions that are made about configuring the Solaris 2.4 system. The following procedure is one way to obtain the disk partition information.

1. Obtain the names of the disks attached to the system.

To obtain the names of the disks attached to the system, use the `format(8)` command, as in the following example.

```
# /usr/etc/format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
  0. sd0 at esp0 slave 24
     sd0: <Quantum ProDrive 105S cyl 974 alt 2 hd 6 sec 35>
  1. sd1 at esp0 slave 8
     sd1: <Quantum ProDrive 105S cyl 974 alt 2 hd 6 sec 35>
Specify disk (enter its number): ^D
#
```

2. Save the disk partition information.

In the previous example, the system is configured with two disks: `sd0` and `sd1`. Now that you know the disks on the system, you can obtain the partition information encoded on each of the disks, using the `dckinfo(8)` command. You can pipe the output to a printer or to a file that you can save to another system as in the following example.

```
# /etc/dckinfo sd0 sd1 | lpr
```

Note – Using the previous command provides you with information only on the configured partitions. All nonconfigured partitions are displayed with the message: “No such device or address.”

Making a List of System Components to Save

Make a list of all the system components in the existing SunOS release 4.x environment and decide which are critical to the user’s system. Consider:

- Locally developed applications
- Any unbundled software products
- Third-party applications
- Third-party peripheral devices and drivers (8 mm tape drives and SBus cards, for example)

Saving File System Information

Make a list of all the locally mounted file systems that you want to transfer to the Solaris 2.5 operating environment.

1. Save file system information.

The mappings between file system names (for example, `/usr`, `/home`) and device names (for example, `/dev/sd0g`) reside in the configuration file `/etc/fstab`. You should make a printed copy of this file before proceeding. This helps you construct the Solaris 2.4 version of the file `/etc/vfstab`.

Make a printed copy of the file system information as in the following example.

```
# lpr /etc/fstab
```

2. Look at the `fstab` file and record all file systems with a third field (the file system type) of 4.2. Here is a sample `/etc/fstab` file.

```
# cat /etc/fstab
/dev/sd0a    /                4.2 rw 1 1
/dev/sd0d    /export          4.2 rw 1 3
/dev/sd1h    /home           4.2 rw 1 4
/dev/sd0g    /usr            4.2 rw 1 2
/dev/sd0e    /export/swap    4.2 rw 1 5
```

Making a List of Files and File Systems to Save

Use the following guidelines to make the list of file systems to save:

- As a general rule, do not transfer file systems containing “system” files (for example, the `/usr` or `/` file systems) in their entirety.
- Do not save temporary file systems, such as `/tmp`.
- Do extract and transfer the data files that have changed locally or those on which the server depends for administrative data such as some `/etc` files (for example, `/etc/hosts`), exported file systems (use the `exportfs` command to list them), and `/tftpboot` directory, which you should save as a safety precaution.
- Do completely preserve file systems containing only locally generated data, such as `spool` and user home directories.
- Be sure to save file systems that contain information about clients if you are migrating a server for SunOS release 4.x clients. Typically, `/export` would be such a file.

Saving Metadevice Configuration Information

Use this section only if you are upgrading a system running the SPARCServer™ Manager or Solstice™ DiskSuite™ unbundled products. (These products are used to mirror, concatenate, or stripe multiple disks.)

If you want to upgrade your system without this product, you will have to modify your multiple-partition configurations to use single partitions. In particular, a concatenated or striped file system must be reorganized onto a single disk, and partitions and mirrors can no longer be used.

If the system is running SPARCserver Manager or Solstice DiskSuite, you should save the metadevice configuration information before installing Solaris 2.4 software. This enables you to recover the state of the metadevices when you install Solaris 2.4 software, and serves as a reference as you construct the list of disks attached to your system.

1. Use the `metastat(8)` command to save information as in the following example.

```
# /etc/metastat -p | lpr
```

2. Save the output of the `metadb(8)` command.
For example.

```
# /etc/metadb -i | lpr
```

The output of `metadb` tells you the state database configuration information. This information is necessary to reconstruct the state databases if you reinstall Solstice DiskSuite.

Deciding the Order of Installation for Networks

If you are migrating a network, decide the order of the systems on which you will install Solaris 2.5 software to maximize convenience for the users. For example, you might want to upgrade all client systems before you upgrade any servers. The first system you upgrade should be a standalone system with a locally attached CD-ROM drive.

For a while, you will probably manage a network consisting of both SunOS release 4.x and Solaris 2.5 systems, and part of your planning should involve determining priorities. For example, you may want to upgrade one domain and use it for system administration testing and for porting internally developed applications before you upgrade the entire network environment.

Backing Up the File Systems

You can use standard commands and procedures given in SunOS release 4.x documentation to do backups. The exact command you use depends on whether the tape drive is local or remote.

To simplify the restore, you should use a separate tape for each file system's backup. If you do place more than one `dump` on a tape, you should include information on the label identifying the file position (sequence number) of each `dump` (for example, 1. /, 2. /usr, 3. /home). The first `dump` on each tape should be numbered 1.

Saving SunOS Release 4.x System Data

After you have backed up all your file systems, make a separate `tar(1)` tape of the system configuration files.

1. Create a file with names of the files, directories, or file systems you want to back up.

You can use the example list on the next page to help select system configuration files you want to save. Make one entry per line in the file with the relative path name in front as in the following example.

```
./etc
./var/spool/mail
./cshrc
./login
.
.
.
```

Note – The list contains suggestions. You should study the items carefully and add to or delete paths depending on the configuration at your site. For example, if you have special files in directories from third-party software vendors, you may need to save them.

If the system is an NIS master server, you should save all the files that reside in the NIS master directory (for example, `/etc`). Additionally, save any other master files that you added to NIS. The suggestions for files to back up include:

- `./cshrc`
- `./profile`
- `./login`
- `./logout`
- `./rhosts`
- `/etc` (if the system is an NIS client or has no name service)
- `/var/spool/calendar`
- `/var/spool/cron`
- `/var/spool/uucp`
- `/var/nis` (if the system is an NIS master server)
- Boot programs in `./tftpboot`

2. Save the system configuration information.

You may want to use a tape rather than a diskette because diskettes are limited to only 1 megabyte of data. Type the following commands to make the tar tape.

```
# cd /
# tar cvf tapedrive -I filename
```

In the previous example, *tapedrive* is the name of the drive where you have inserted the tape for the `tar` command (for example, `/dev/rst0`). *filename* is the name of the file you created in step 1 of this section.

Installing Solaris Software

Install Solaris 2.5 software on the server or standalone system using software installation procedures given in *SPARC: Installing Solaris Software*.

Note – SunOS release 5.x does not support NIS. Clients that use the NIS or DNS naming service must either use the NIS binary compatibility package or continue to use a SunOS release 4.x NIS or DNS server.

Restoring Data

This section describes issues related to restoring the user and system data you backed up before installing the Solaris 2.5 software. If the system will serve both Solaris 2.5 clients and SunOS release 4.x clients, also see Chapter 10, “Setting Up a Solaris 2.5 Server to Support SunOS Release 4.x Diskless/Dataless Clients.”

Restoring SunOS Release 4.x User Data

It is relatively easy to convert most SunOS release 4.x user data to run under a SunOS release 5.5 system. Many shell scripts will work as they are, although you should look at the next subsection for guidelines on restoring them.

Binary executable files should be recompiled for maximum efficiency. If they cannot be recompiled, they may run in binary compatibility mode if the Binary Compatibility Package is installed.

The names of the tape drives are different in Solaris 2.5. For instance, `/dev/nrst0` is now `/dev/rmt/0n`. In addition, the `restore(8)` command is now named `ufsrestore(1M)` and resides in `/usr/sbin/ufsrestore`. See “Device Naming Conventions” on page 61 if you need help with Solaris 2.5 device names.

Converting the Environment Initialization Files

The SunOS release 4.1 environment initialization files, such as `.login`, `.cshrc`, and `.profile`, can be copied directly to the SunOS release 5.3 system software to set up the working environment. However, some SunOS release 4.x commands have been renamed or moved in SunOS release 5.5. Before running them, examine the files and replace any incompatible SunOS release 4.x commands with their SunOS release 5.5 counterparts. Appendix A, “Commands Reference Table,” shows which SunOS release 4.x commands are no longer available.

To use the SunOS release 4.x initialization files to set up the SunOS release 5.5 work environment, select the same login shell in the SunOS release 5.5 environment that was used in SunOS release 4.x.

Restoring File Systems From Separate Tapes

If you created a separate dump tape for each file system containing data, run the following commands for each file system you want to restore.

```
# /usr/sbin/newfs device_name
# mkdir -p filesystem_name
# /etc/mount -F ufs device_name file_system_name
# cd filesystem_name
# /usr/sbin/ufsrestore rf tape_device_name
```

In the previous command sequence, the *device_name* is the name of the disk slice where the file system resides (for example, `/dev/dsk/c0t3d0s7`, which was `/dev/sd0h` under SunOS release 4.x). See “Device Naming Conventions” on page 61 for help understanding device names. *filesystem* is the mount point for the file system you are restoring (for example, `/export/home`).

Note – Check in advance to ensure that the target slice is large enough to accommodate the file system being restored.

Restoring File Systems From a Single Tape

If you did not create separate dump tapes for each file system, you can use a similar command sequence to that listed previously, but you need to position the tape to the tape file mark corresponding to the dump you want to restore. Do this using the following command.

```
# mt -f tape_device_name fsf count
```

Proceed with the steps listed in the last section. See the `mt(1M)` man page for additional information.

Note – After the user data is restored on the Solaris 2.4 system, you may need to merge the `/etc/fstab` information from the SunOS release 4.x system into the SunOS release 5.5 `/etc/vfstab` so it is automatically checked and mounted at boot time. The following subsection explains how.

Restoring SunOS Release 4.x System Data

This section describes how to restore the archive of system data created using instructions from the first part of this chapter. First, you must restore the information to a temporary directory on the Solaris 2.4 system.

Use the following commands to restore system data to a temporary directory (called `old` in the example).

```
# mkdir /var/tmp/old
# cd /var/tmp/old
# tar xvf device_name
```

After the information is back on the system in the temporary directory, you need to make it available in the Solaris 2.5 operating environment. Some of the data can just be merged into the files, while some types of data must be converted to new formats. This subsection gives examples of both types of data.

The system's configuration defines which files you need to work with. Complete the restore by merging or converting files as follows:

- **Systems with no name service:**
If the system has no name service, merge or convert all the relevant system files located in `/etc` and `/var`.
- **Systems that are NIS clients:**
If the system is an NIS client, merge or convert only the local system configuration files located in `/etc` and `/var` that are not provided via the NIS name service.
- **Systems that are NIS master servers:**
If the system is an NIS master server, merge or convert all the files that reside in the NIS master directory (for example, `/etc`). Additionally, update other local configuration files in `/etc` and `/var`.

Files to Merge

To make data from any of the following files available, merge the changes into the Solaris 2.4 version of the same file. Note, however, that not all of these files were modified on the SunOS 4.x system. Identify files that were changed on

the SunOS release 4.x system and merge these only. As you read the list, note that some of the file names are slightly different. For example, `/etc/auto.*` are now `/etc/auto_*`.

The following is an example list of the SunOS release 4.x files backed up using the instructions in the first part of this chapter. These files are candidates for merging into the Solaris 2.4 operating environment.

- All automounter maps, including `/etc/auto.master` and any others
- `/etc/aliases`
- `/etc/bootparams`
- `/etc/ethers`
- `/etc/hosts`
- `/etc/format.dat`
- `/etc/inetd.conf`
- `/etc/netmasks`
- `/etc/networks`
- `/etc/protocols`
- `/etc/publickey`
- `/etc/rpc`
- `/etc/services`
- `/etc/hosts.equiv`
- `/etc/remote`
- `/.cshrc`
- `/.profile`
- `/.login`
- `/.logout`
- `/.rhosts`
- `/var/spool/cron`
- `/var/spool/mail`
- `/var/spool/calendar`
- `/var/spool/uucp`

Files to Convert

Many system files, such as the `/etc/fstab` file, have been replaced and do not exist under the Solaris 2.4 operating environment. Information from these files must be extracted and manually converted in the Solaris 2.5 environment. For example, `/etc/printcap` entries must be typed into the Solaris 2.5 printer subsystem (see the `lpadmin (1M)` man page). See Appendix D, “System Files Reference Table,” to examine SunOS release 4.x files for changes.

Do not restore operating system executable files (such as system commands in `/usr/bin`) from the SunOS release 4.x system to your system after installing the Solaris 2.5 software.

You must change the following files before merging the data onto the Solaris 2.5 system.

- `/etc/uucp` – There have been some changes to the UUCP system. The `Config`, `Grades`, and `Limits` files are new in the Solaris 2.4 operating environment. The files `Devconfig`, `Devices`, `Dialcodes`, `Dialers`, `Permissions`, `Poll`, `Sysfiles`, and `systems` are the same in the Solaris 2.4 operating environment as they were in SunOS release 4.x. These files can be merged together. There are also several SunOS release 4.x files that are not used in the Solaris 2.4 operating environment.
- `/etc/group` – The basic format of this file is the same as it was in the SunOS 4.1 and SunOS 4.1.x releases. However, previous releases used a group entry beginning with a plus sign (+) or minus sign (-) to selectively incorporate entries from NIS maps for group. See the `group(4)` man page if that compatibility is needed under the Solaris 2.4 operating environment.
- `/etc/netgroup` – There is no `/etc/netgroup` file in SunOS release 5.5.
- `/etc/exports` – File systems to be shared on the network under the Solaris 2.4 operating environment use the `/etc/dfs/dfstab` file instead of `/etc/exports`. The format of entries in this file follows.

```
share -F fstype -o options -d "text" pathname resource
```

See the `dfstab(4)` man page for additional information.

- `/etc/fstab` – File systems to be mounted under the Solaris 2.4 operating environment use the `/etc/vfstab` file instead of `/etc/fstab`. The format of entries in the `/etc/vfstab` file follows.

```
dev raw_dev mnt_pt fs_type fsck_pass auto_mnt mnt_option
```

Refer to the `vfstab(4)` man page for additional information.

- `/etc/passwd` – The format of the `passwd` file is the same as that under SunOS release 4.x. However, user passwords are now stored in the `/etc/shadow` file. Refer to the `passwd(4)` and the `shadow(4)` man pages for additional information.

-
- `/etc/sendmail.cf` – The format of `sendmail.cf` is the same as that under SunOS release 4.x. The location of the file is now `/etc/mail/sendmail.cf`.
 - `/etc/ttytab` – Under SunOS release 4.x, `ttytab` was used to control serial ports and the characteristics of the terminals on those serial lines. Under the Solaris 2.4 operating environment, the Service Access Facility is used to configure this capability.
 - `/etc/printcap` – Under the Solaris 2.4 operating environment, printers are configured using the System V Release 4 print services. See *System Administration Guide, Volume II* for additional information.

Using the Compatibility Packages



SunOS release 5.5 is neither source nor binary compatible with SunOS release 4.x. This means that SunOS release 4.x programs and user applications based on those releases may not run correctly under the Solaris 2.5 operating environment. Compatibility packages make it possible for these programs to run on a Solaris 2.5 system.

This chapter briefly discusses two compatibility packages: the SunOS/BSD Source Compatibility Package and the Binary Compatibility Package. These packages make the transition easier by enabling you to use SunOS release 4.x commands and applications while your environment and applications migrate to the Solaris 2.5 operating environment.

This chapter includes the following sections:

<i>Why Port Applications?</i>	<i>page 40</i>
<i>SunOS/BSD Source Compatibility Package</i>	<i>page 40</i>
<i>Binary Compatibility Package</i>	<i>page 40</i>

Some SunOS release 4.x commands are not available in the Solaris 2.5 operating environment. Others exist, but have changed. For information about changes to SunOS release 4.x commands in the Solaris 2.5 operating environment, see Appendix A, “Commands Reference Table.”

Why Port Applications?

Although the SunOS Binary Compatibility Package and the SunOS/BSD Source Compatibility Package allow you to use applications as they are, you should port applications as soon as possible. Long-term reliance on the compatibility packages is not advised for the following reasons:

- The application's performance is reduced.
- You will not be able to take advantage of the Solaris 2.5 operating environment's increased range of operations and portability.
- Compatibility packages are temporary aids to help sites through the transition.

SunOS/BSD Source Compatibility Package

The SunOS BSD/Source Compatibility Package is an optional package available with the Solaris 2.5 operating environment. The package contains a collection of SunOS release 4.x and BSD commands, library routines, and header files otherwise not available with the Solaris 2.5 operating environment. The Binary Compatibility Package must be installed in order to use the SunOS/BSD Source Compatibility Package.

The interfaces in the SunOS/BSD Source Compatibility Package are installed in the `/usr/ucb` directory, thereby avoiding conflicts with existing SunOS release 5.5 interfaces. These interfaces provide a familiar SunOS environment while your environment and applications are migrating to SunOS release 5.5. To use these interfaces, you must either specify the full path name, or modify your `PATH` environment variable. When modifying your `PATH` environment variable, note that `/usr/ucb` should precede `/usr/bin`.

For detailed information about the Source Compatibility Package, see *Source Compatibility Guide*.

Binary Compatibility Package

The Binary Compatibility Package is an optional package available with the Solaris 2.5 operating environment. The package allows existing SunOS release 4.x applications, both statically and dynamically linked, to run under the Solaris 2.5 operating environment without modification or recompilation. It

handles most binary interface discrepancies between the two releases transparently. This results in a Solaris 2.5 operating environment where SunOS release 4.x applications can run properly.

See *Binary Compatibility Guide* for procedures about setting up your environment to access this package. This guide also details the limitations of the Binary Compatibility Package.

Using the Binary Compatibility Package to Run SunOS Release 4.x Applications

The Binary Compatibility Package allows most applications to run under the Solaris 2.5 operating environment, making them available for use before they are ported to SunOS release 5.5. With this package, well-behaved application binaries based on SunOS release 4.x system software will run under SunOS release 5.5 without modifications or recompilation.

The Binary Compatibility Package is intended for end-user environments, not for use as a development environment. All SunOS release 5.5 application development should be done under the base SunOS release 5.5 environment.

Security for the Solaris 2.5 operating environment combines several features from SunOS release 4.x and AT&T SVR4 with capabilities added specifically for Solaris 2.5. There are also changes in the packaging of some SunOS release 4.x security programs.

This chapter describes major differences between SunOS release 4.x and Solaris 2.5 operating environment security, and points out how those changes may affect system administration procedures. *System Administration Guide, Volume II* describes the administration and use of these features more fully.

This chapter contains the following sections:

<i>Solaris 2.5 Security Features</i>	<i>page 44</i>
<i>Password Aging Changes</i>	<i>page 45</i>
<i>Automated Security Enhancement Tool (ASET)</i>	<i>page 46</i>
<i>Unbundled Security Options</i>	<i>page 47</i>

Solaris 2.5 Security Features

Most of the security features from SunOS release 4.x systems are also available in the Solaris 2.5 operating environment. These include:

- Internet security
- `.rhosts` and `.rhosts.equiv` files
- Secure RPC and NFS

NFS Administration Guide documents secure NFS and the `.rhosts` files. *TCP/IP and Data Communications Administration Guide* describes administering Internet security.

Security for local SunOS release 5.5 systems includes storing encrypted passwords in a separate file, controlling login defaults, and restricted shells. Equivalent NIS+ security, described in *NIS+ and FNS Administration Guide*, controls network-wide access to systems.

The following subsections summarize security features under local system control.

/etc/passwd and /etc/shadow Files

The SunOS release 5.5 `passwd` command stores encrypted versions of passwords in a separate file, `/etc/shadow`, and allows only root access it. This prevents general access to the encrypted passwords that formerly appeared in the `/etc/passwd` file, which anyone could read.

The `/etc/shadow` file also includes entries that force password aging for individual user login accounts. The mechanism for changing entries to the `passwd` and `shadow` files is described in *System Administration Guide, Volume I*.

`/etc/default` *Files*

Several files that control default system access are stored in the `/etc/default` directory. These files limit access to specific systems on a network. Table 5-1 summarizes the files in the `/etc/default` directory.

Table 5-1 Files in `/etc/default` Directory

<code>/etc/default/login</code>	Controls system login policies, including root access. The default is to limit root access to the console.
<code>/etc/default/passwd</code>	Controls default policy on password aging
<code>/etc/default/su</code>	Controls which root (<code>su</code>) access to system will be logged and where it will be displayed

Restricted Shells

System administrators can use restricted versions of the Korn shell (`rksh`) and Bourne shell (`rsh`) to limit the operations allowed for a particular user account.

Restricted shells do not allow the following operations:

- Changing directories
- Setting the `$PATH` variable
- Specifying path or command names beginning with “/”
- Redirecting output

See the `ksh` and `sh` man pages in the *man Pages(1): User Commands* for a description of these shells.

Note that the restricted shell and the remote shell have the same command name (`rsh`) with different path names:

- `/usr/lib/rsh` is the restricted shell
- `/usr/bin/rsh` is the remote shell

Password Aging Changes

The SunOS release 5.5 system features password aging. This feature assigns a limited lifetime to each user password to maintain password secrecy. As a password reaches the end of its life, the password owner is notified and prompted to select a new one.

You can implement password aging using one of the following methods:

- *Method 1* – Use Admintool to manage users if you are running OpenWindows. For information about this method, see *Solaris Advanced User's Guide*.
- *Method 2* – Use new `passwd` or `nispasswd` command options (depending on which name service stores the account).

A system administrator can also set up password aging.

You can change a user password in one of two ways:

- *Method 1*– Use either `passwd` or `nispasswd`, depending on which name service is used to store your account.
- *Method 2* – Use Admintool to manage users if you are running OpenWindows. For information about this method, see *Solaris Advanced User's Guide*.

For more information on `passwd` and `nispasswd`, see the command tables in Appendix D, “System Files Reference Table.”

Automated Security Enhancement Tool (ASET)

ASET is included with the Solaris 2.5 operating environment. It was available as a separate option with SunOS release 4.x systems. ASET allows you to specify an overall system security level (low, medium, or high) and automatically maintain systems at those levels. This tool can be set up to run on a server and all its clients or on individual clients.

ASET performs these tasks:

- Verifies system file permissions
- Verifies system file contents
- Checks integrity of group file entries
- Checks system configuration files
- Checks environment files (`.profile`, `.login`, and `.cshrc`)
- Verifies EEPROM settings to restrict console login access
- Allows establishment of a firewall or gateway system

System Administration Guide, Volume II describes ASET setup and monitoring in detail.

Unbundled Security Options

Currently available unbundled security options are Kerberos security and SunShield™.

Kerberos Security

The Solaris 2.5 operating environment includes support for Kerberos authentication for secure RPC. Kerberos source code and administrative utilities are available from MIT.

Solaris 2.5 operating environment Kerberos support includes:

- Client applications library that can use Kerberos
- Kerberos option to Secure RPC
- Sun's NFS® distributed computing file system application with Kerberos
- Commands to administer user tickets on the client

System Interfaces Guide describes the use of Kerberos with Secure RPC. *NFS Administration Guide* describes the use of Kerberos with NFS.

SunShield

Password hiding and aging were always available; however, with the Solaris 2.5 operating environment, it was necessary to add separate software, SunShield ARM, to move encrypted passwords out of `/etc/passwd` to `/etc/shadow`, and to ask users to change their passwords periodically.

User Environment Administration



This chapter describes differences in tasks you may perform to set up the local user environment after installing the Solaris 2.5 software.

<i>Selecting a Default Shell</i>	<i>page 49</i>
<i>Customizing User Environments</i>	<i>page 51</i>
<i>Window System</i>	<i>page 52</i>
<i>User and Group Administration</i>	<i>page 53</i>
<i>Using Mail</i>	<i>page 54</i>
<i>Using Document Tools</i>	<i>page 55</i>
<i>man Page Organization Differences</i>	<i>page 56</i>

Selecting a Default Shell

The login shell is the command interpreter that runs when you are logged in. The Solaris 2.5 operating environment offers three shells:

- Bourne shell, the default shell (/bin/sh)
- C shell (/bin/csh)
- Korn shell (/bin/ksh)

If you use the shell often, you may prefer to use the C shell or the Korn shell because of their interactive capabilities. Table 6-1 lists the features of all three shells.

Table 6-1 Basic Features of the Bourne, C, and Korn Shells

Feature	Bourne	C	Korn
Syntax compatible with <code>sh</code>	Yes	No	Yes
Job control	Yes	Yes	Yes
History list	No	Yes	Yes
Command-line editing	No	Yes	Yes
Aliases	No	Yes	Yes
Single-character abbreviation for login directory	No	Yes	Yes
Protect files from overwriting (<code>noclobber</code>)	No	Yes	Yes
Ignore Control-D (<code>ignoreeof</code>)	No	Yes	Yes
Enhanced <code>cd</code>	No	Yes	Yes
Initialization file separate from <code>.profile</code>	No	Yes	Yes
Logout file	No	Yes	No

Use one of the following methods if you want to change from one shell to another:

- *Method 1* – Edit the information in the last field of the line in the `/etc/passwd` file that begins with your login name. If this entry is blank or `sh`, the login shell is the Bourne shell; if the entry is `csh`, the login shell is the C shell; if the entry is `ksh`, the login shell is the Korn shell.
- *Method 2* – In OpenWindows, use Admintool. See *Solaris Advanced User's Guide* for information.

After you change to a new shell, log out and log in again, to activate the shell.

Customizing User Environments

This section describes how to determine which initialization files to edit to customize the local environment based on your choice of login shell, and where to find them in the SunOS release 5.5 file systems. Set up your environment by editing the variables in the initialization files. The default shell determines which files you need to edit: `.profile`, `.login`, or `.cshrc`. Table 6-2 shows the initialization files for the Bourne, C, and Korn shells.

Table 6-2 Initialization Files for Bourne, C, and Korn Shells

Shell	Initialization File	Purpose
Bourne	<code>/etc/profile</code>	Defines system profile at login
	<code>\$HOME/.profile</code>	Defines user's profile at login
C	<code>/etc/.login</code>	Defines system environment at login
	<code>\$HOME/.cshrc</code>	Defines user's environment at login
	<code>\$HOME/.login</code>	Defines user's profile at login
Korn	<code>/etc/profile</code>	Defines system profile at login
	<code>\$HOME/.profile</code>	Defines user's profile at login
	<code>\$HOME/ksh_env</code>	Defines user's environment at login in the file specified by the <code>ksh_env</code> variable

In this release, the shell initialization-file templates have moved to the `/etc/skel` directory from `/usr/lib`, where they were in SunOS release 4.x. The template file locations are shown in Table 6-3. Copy the template file (or files) for the appropriate default shell to your home directory before you modify it.

Table 6-3 Default Home Directory Startup Files

Shell	File
Bourne	<code>/etc/skel/local.profile</code>
C	<code>/etc/skel/local.login</code>
	<code>/etc/skel/local.cshrc</code>
Korn	<code>/etc/skel/local.profile</code>

For information on setting up initialization files, see *System Administration Guide, Volume I*.

Using the SunOS 4.x Work Environment With Solaris

If you prefer to continue using the SunOS release 4.x work environment, SunOS release 5.5 can use the old system files and initialization files such as `.login`, `.cshrc`, and `.profile` to re-create the look and feel of SunOS release 4.x. Many of these SunOS release 4.x files can be converted, or used as they are, and executed easily.

The installation process in Chapter 3, “Installation and Configuration Changes,” explains how to re-create the SunOS release 4.x environment within the Solaris 2.5 operating environment.

Window System

OpenWindows 3.2 is the default Solaris 2.5 windowing environment. The OpenWindows environment offers a simple and intuitive interface for users of Solaris 2.5 system software. If you have been using the OpenWindows 2.0 environment, you will notice that the OpenWindows 3.1 icons have changed and some applications are not compatible with the OpenWindows 3.1 platform.

The OpenWindows Developer’s Guide File Chooser (`gfm`) regular expression file pattern matching code (`filter_pat`) is slightly different from the regular expression file pattern matching code in the XView File Chooser object. This could result in the same regular expression matching slightly different sets of files in the two different choosers. The XView File Chooser uses `/usr/include/reexp.h` in SunOS release 5.5 and its usage is correct.

SunView™ software is not part of the Solaris 2.5 operating environment. If you have been using the SunView environment, you will find that your SunView applications are incompatible with the OpenWindows environment and must be converted.

See *OpenWindows Version 3.1 User’s Guide* for information about:

- Features of OpenWindows 3.1
- The applications that are not compatible between OpenWindows Version 2.0 and 3.1 platforms
- Guidelines for modifying incompatible applications

User and Group Administration

This section describes your options for performing user and group administration.

User and Group Administration Choices

You can add, modify, and remove users and groups through the command-line interface using `useradd`, `userdel`, and `usermod`. Although these commands are not as robust as Admintool, they do allow you to do most of the tasks supported by Admintool from the command line without running OpenWindows.

The `useradd`, `userdel`, and `usermod` commands are similar to editing the `/etc` files in that they also affect only the local system. These commands cannot be used to change any information in the network naming service. However, you can use `useradd` to verify the uniqueness of the user name and user ID and the existence of group names in the network naming service.

Adding User Accounts

This section describes changes to the general procedure for adding user accounts.

Changes to the Procedure

The general procedure for adding new users to a SunOS release 4.x system was:

1. Edit the `/etc/passwd` file and add an entry for the new user.
2. Create a home directory and set the permissions for the new user.
3. Set up skeletal files for the new user (`.cshrc`, `.login`, `.profile`...).
4. Add the new user to the naming service (NIS).

In the Solaris 2.5 operating environment, there are three ways to add (and maintain) user accounts:

- Use Admintool – This is the most straight forward method to use if the system is running the OpenWindows environment.
- Use command-line interfaces (`useradd`, `usermod`, and `userdel`) – Use this method if you don't want to use Admintool.

- Manually edit files (similar to the SunOS release 4.x procedure with a few exceptions)

Note – Because SunOS release 5.5 uses a shadow password file, it is no longer sufficient just to edit the `/etc/passwd` file. You should not attempt this method unless you have ample experience with this type of administration.

System Administration Guide, Volume I describes in detail the policy decisions you should consider before you begin to set up accounts. *System Administration Guide, Volume II* explains security considerations for controlling user access to systems and networks.

Using Mail

The SunOS release 4.x mail programs are different in the Solaris 2.5 operating environment; however, procedures for setting up mail are still the same. The SunOS release 4.x version of `mail` is included in the SunOS/BSD Source Compatibility Package. Its user interface is different from the Solaris 2.5 operating environment's version of `mail`. Additionally, some useful mail facilities are included for compatibility.

In the Solaris 2.5 operating environment, there are three programs for sending and retrieving your mail. All three are backward compatible and can be used to read your old SunOS release 4.x mail. They are:

- `mailtool`, the OpenWindows interface for the mail program. New Solaris 2.5 `mailtool` options enable you to attach files to your messages, include third-party messages with your mail, deliver mail to multiple recipients, and send audio messages.

See *OpenWindows Version 3.1 User's Guide* for a complete discussion of `mailtool`.

- `mailx`, which is installed under `/usr/bin/mailx`. This is the Solaris 2.5 mail reading program. It is an enhanced version of SunOS release 4.x `/usr/ucb/mail`. In the Solaris 2.5 operating environment, `/usr/ucb/mail` is a link to `/usr/bin/mailx`. `mailx` offers message headers that allow you to preview the sender and subject of each message before you read them. You can also switch between reading, sending, and editing mail messages.

See the `mailx(1)` man page for more information on `mailx`.

- `mail` refers to the mail program under `/usr/bin/mail`. The Solaris 2.5 interface is similar to the SunOS release 4.x `/usr/bin/mail` version (see the `bin-mail(1)` manual page in *SunOS 4.x Reference Manual*).

See the `mail(1)` man page for more information on `mail`.

For a complete discussion of all Solaris 2.5 mail programs, see *Mail Administration Guide*.

Using Document Tools

This section outlines the main differences in using document tools between SunOS release 4.x and the Solaris 2.5 operating environment.

- The Solaris 2.5 operating environment provides a set of PostScript® filters and device-independent fonts. However, most SunOS release 4.x TranScript® filters have SunOS release 5.5 equivalents, while a few less common ones do not. In SunOS release 5.5 systems, there is no T_EX filter, no `pscat` (C/A/T) filter, and no raster image filter.

- The Solaris 2.5 operating environment provides device-independent `troff`, with the following features: SunOS release 4.x `troff` input files work with Solaris 2.5 `troff`; `troff` default output goes to the standard output instead of the printer. Therefore, you must specify a printer when you send `troff` output to the printer.

man *Page Organization Differences*

man pages organization has changed to be compatible with SVR4 organization. As a result, some sections have been renamed. For example, man (8) is now man (1M).

Table 6-4 shows SunOS release 5.5 man page directories.

Table 6-4 SunOS Release 5.5 man Page Directories

/man		
Directory	Contents	Suffixes
man1	User commands	1B - SunOS/BSD compatibility commands 1C - Communication commands 1F - FMLI commands 1S - SunOS commands
man1M	System administration commands	
man2	System calls	
man3	Library functions	3B - SunOS/BSD compatibility libraries 3C - C library functions 3E - ELF library functions 3G - C library functions 3I - Wide Character functions 3K - Kernel VM library functions 3M - Math library 3N - Network functions 3R - RPC services library

Table 6-4 SunOS Release 5.5 man Page Directories (Continued)

Directory	Contents	Suffixes
/man		3S - Standard I/O functions 3T - Threads library functions 3X - Miscellaneous library functions
man4	File formats	4B - SunOS/BSD compatibility file formats
man5	Headers, tables, and Macros	
man7	special files	
man9	DDI/DKI	
man9E	DDI/DKI entry points	
man9F	DDI/DKI kernel functions	
man9S	DDI/DKI data structures	

Customizing the man Command Search Path

Unlike SunOS release 4.x, which searched the individual `man` directories according to a predetermined order, SunOS release 5.5 lets you determine the search path. The `man` command uses the path set in the `man` page configuration file, `man.cf`.

Each component of the `MANPATH` environment variable can contain a different `man.cf` file. You can modify `man.cf` to change the order of the search; for example, to search 3b before 3c. The configuration file for the `/usr/share/man` directory follows.

```
#
# Default configuration file for the on-line manual pages.
#
MANSECTS=1,1m,1c,1f,1s,1b,2,3,3c,3s,3x,3i,3t,3r,3n,3m,3k,3g, \
3e,3b,9f,9s,9e,9,4,5,7,4b,6,1,n
```

The arguments to MANSECTS are derived from the man subdirectories available. The number of subdirectories has increased dramatically in this release because each subsection has its own directory. This new structure improves the performance of the man command and gives you finer control over the search path. The next two figures compare the man directories for the two releases.

```

sunos4.1% ls /usr/share/man
man1/  man2/  man3/  man4/  man5/  man6/  man7/  man8/
man1/  mann/

```

```

sunos5.3% ls /usr/share/man
man.cf  man1f/  man3/  man3g/  man3n/  man3x/  man6/  man9f/
man1/   man1m/  man3b/  man3i/  man3r/  man4/   man7/  man9s/
man1b/  man1s/  man3c/  man3k/  man3s/  man4b/  man9/  man1/
man1c/  man2/   man3e/  man3m/  man3t/  man5/   man9e/  mann/

```

whatis and windex *Databases*

In SunOS release 4.x, the man page table of contents and keyword database is called *whatis*. In SunOS release 5.5, this information is in the *windex* file. In both releases, the database is created by the *catman* command, and is used by the *man*, *apropos*, and *whatis* commands.

The *windex* file also has a slightly different format than the *whatis* file, as you can see from the following comparison of the two release versions.

```

sunos4.1% man -k tset
tset, reset (1)  - establish or restore terminal characteristics

```

```

sunos5.3% man -k tset
reset tset (1b)  - establish or restore terminal characteristics
tset  tset (1b)  - establish or restore terminal characteristics

```

Using the `man` Command

Table 6-5 shows that SunOS release 5.5's version of the `man` command has additional search options.

Table 6-5 New `man` Command Options

Option	Description
<code>-a</code>	Displays all <code>man</code> pages that match <i>file name</i> . The pages are displayed sequentially in the order they are found.
<code>-l</code>	Lists all <code>man</code> pages that match the <code>file name</code> . You can use the output of this command to specify a section number with the <code>-s</code> option.
<code>-s section-number</code>	Searches <i>section-number</i> for <i>file name</i> . In SunOS release 4.x, the <code>man</code> command accepted the section number as an option; in this release, the section number must be preceded by <code>-s</code> .
<code>-F</code>	Forces the <code>man</code> command to search all directories until <i>file name</i> is found. This option overrides the <code>windex</code> database and the <code>man.cf</code> file.

See the `man(1)` `man` page for a complete description of the SunOS release 5.5 `man` command.

Device Administration



This chapter explains SunOS release 5.5 device naming conventions and discusses changes to device-related tasks such as getting information about disks, adding devices to a system, and using the Volume Management.

This chapter contains the following sections:

<i>Device Naming Conventions</i>	<i>page 61</i>
<i>Obtaining Disk Information</i>	<i>page 63</i>
<i>Adding Devices to the System</i>	<i>page 65</i>
<i>Using Volume Management</i>	<i>page 66</i>

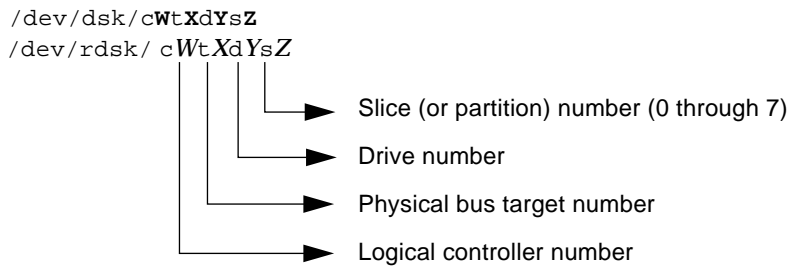
Device Naming Conventions

Device naming conventions have changed between SunOS release 4.x and SunOS release 5.5. In addition, the `/dev` directory, which contains the special device names, has been changed from a flat directory to a hierarchical one, with a separate subdirectory for each category of device. For example, the location of disk device files is `/dev/dsk`, while raw disks are located in `/dev/rdsk`.

SunOS release 5.5 commands that take device names as arguments must use the SunOS release 5.5 device naming conventions. However, you can still use and recognize the SunOS release 4.x device names if you install the SunOS/BSD Source Compatibility Package. See *Source Compatibility Guide* for additional information.

Convention for Disks

The disk partition slice numbers (0 through 7) correspond to partitions a through h of previous SunOS releases.



Note – Most SCSI disks have embedded controllers. This means that the drive number will always be “0”, but the target number varies. For example, if an external disk drive has its rear switch set to “2”, the device name for the first slice is `/dev/dsk/c02d0s0`, not `/dev/dsk/c0t0d2s0`.

Because the names for SCSI targets 0 and 3 were reversed on some Sun4c systems, device naming can be confusing. Under SunOS 4.1.x, SCSI target 3 was called `sd0`, but is now properly named `c0t3d0`. SCSI target 0 was called `sd3`, but is now named `c0t0d0`. Other SCSI disk names translate normally. For example, in SunOS release 5.5 `sd2a` is `c0t2d0s0` and `sd2b` is `c0t2d0s1`.

Convention for Tape Drives

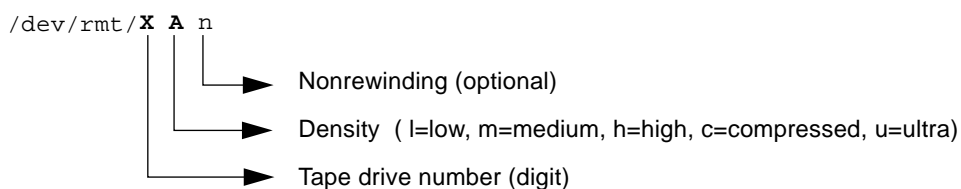


Table 7-1 provides some examples that compare the SunOS release 4.x and SunOS release 5.5 device naming conventions.

Table 7-1 SunOS Release 4.x and SunOS Release 5.5 Device Names

Device Description	SunOS release 4.x Device Name	SunOS release 5.5 Device Name
Disk Devices	<code>/dev/sd0g</code>	<code>/dev/dsk/c0t3d0s6</code>
	<code>/dev/rsd3b</code>	<code>/dev/rdisk/c0t0d0s1</code>
	<code>/dev/rsd3a</code>	<code>/dev/rdisk/c0t0d0s0</code>
Magnetic Tape Devices	<code>/dev/nrmt8</code>	<code>/dev/rmt/8hn</code>
	<code>/dev/rst0</code>	<code>/dev/rmt/0</code>
CD-ROM Device	<code>/dev/sr0</code>	<code>/dev/dsk/c0t6d0s2</code>

Obtaining Disk Information

The commands that report disk information in the SunOS release 5.5 release have changed. `df(1M)`, `du(1M)` are still available, but have changed. `dkinfo(8)`, and `devinfo(1M)` are replaced by `prtvtoc` and `sysdef -d`. This section provides an overview of those changes.

If you have installed the compatibility packages, SunOS release 4.x command versions can be found under `/usr/ucb/df` and `/usr/ucb/du`.

`df` *Command*

The `df` command has been changed to support the VFS architecture. As with the other VFS commands, there are generic and file system versions of the command. The syntax in the SunOS release 5.5 command differs significantly from that used in the SunOS release 4.x version (see Appendix A, “Commands Reference Table,” for more information).

The `df` command now reports disk space in 512-byte blocks instead of kilobytes, but the `-k` option can be used to report disk space in kilobytes. Also, the `-t` option behaves differently; formerly, it restricted the output to file systems of a specified type (for example, `nfs` or `4.2`). The SunOS release 5.5 version produces a full listing with totals.

Finally, use the SunOS release 5.5 device naming conventions when specifying special device names to this command. See “Device Naming Conventions” on page 61 for details.

`du` *Command*

Like `df`, the `du` command reports disk usage in 512-byte blocks instead of kilobytes. There’s also a `-r` option that causes the normally “silent” command to generate messages when it has difficulty reading a directory or opening a file.

`dkinfo` *Command*

The SunOS release 4.x `dkinfo` command is no longer available. To print device information, use `prtvtoc(1M)` instead of `dkinfo`.

The `prtvtoc` command reports the important information stored on a disk’s label, including information on the disk’s partitions. For more information about `prtvtoc`, see *System Administration Guide, Volume I*.

Figure 7-1 shows output for the SunOS release 5.5 `prtvtoc` command.

```
# prtvtoc /dev/rdisk/c0t2d0s2
* /dev/rdisk/c0t2d0s2 partition map
*
* Dimensions:
*   512 bytes/sector
*   36 sectors/track
*   9 tracks/cylinder
*   324 sectors/cylinder
*   1272 cylinders
*   1254 accessible cylinders
*
* Flags:
*   1: unmountable
*   10: read-only
*
*
* Partition  Tag  Flags      First      Sector  Last
* Directory  Tag  Flags      Sector    Count   Sector  Mount
0           0    00          0        32724   32723   /
1           0    00       32724        65448   98171
2           0    00          0       406296  406295
6           0    00       98172       308124  406295  /usr
```

Figure 7-1 SunOS release 5.5 `prtvtoc` Command Output

`devinfo` Command

The SunOS release 4.x version of `devinfo` is incompatible with the SunOS release 5.5 version. To produce output similar to the SunOS release 4.x version, use `prtconf` with the `-v` option.

Adding Devices to the System

At boot time, the system does a self-test and checks for all devices that are attached to it. After you add a new device to the system, use `boot -r` to activate dynamic reconfiguration of the kernel. A reconfiguration script is run to load all the device drivers listed in the module's directories and to create the corresponding hardware nodes. See the `kernel(1M)` man page for more information.

You can also use `boot -a` to interactively add drivers or modules to the system, but you should be aware that if you do, you will be asked to provide other boot parameters, including what to boot and where the root file system is.

Paths to the system files and kernel modules are stored in `/etc/system`. When the system boots, it reads the information in `/etc/system` to determine which modules to load. You can specify a different path by using the `MODDIR` syntax of the `system(4)` file or by using `boot -a`.

For more information about `boot(1m)` or about adding devices and drivers, see *System Administration Guide, Volume I*.

Using Volume Management

Beginning with the Solaris 2.2 software, a new layer of software manages CD-ROM and diskette devices — Volume Management. This software automates the interaction between you and your CDs and diskettes.

OpenWindows users will notice a slight change to the File Manager — it has been modified to use Volume Management to provide immediate user access to CDs and diskettes with file systems. See *Solaris User's Guide* for more information on File Manager's new features.

There are also several new commands to help you administer Volume Management on your system.

Volume Management automatically mounts the CD and diskette file systems when removable media is inserted into the devices. If the CD or diskette contains a file system, it will be automatically mounted in the locations shown in Table 7-2.

Table 7-2 Location of CD-ROM and a Diskette With a File System

Media	Location
CD	<code>/cdrom/cdrom_name</code>
Diskette	<code>/floppy/floppy_name</code>

If the CD or diskette does not contain a file system, it will be accessible in the following locations shown in Table 7-3.

Table 7-3 Location of a CD-ROM and a Diskette Without a File System

Media	Location
CD	/vol/dev/aliases/cdrom0
Diskette	/vol/dev/aliases/floppy0

For security reasons, these file systems are mounted `setuid`. See the `mount(1M)` man page for a description of this and other mount options.

For more information on configuring Volume Management and on using diskettes and CD-ROMs, see *System Administration Guide, Volume I*.

Manual pages for Volume Management components are also available, see `rmmount(1)`, `rmmount.conf(4)`, `volcancel(1)`, `volcheck(1)`, `vold(1M)`, `volmgt(3)`, `vold.conf(4)`, `volfs(7)`, and `volmissing(1)`.

Note – Volume Management now controls these CD-ROM paths:

```
/dev/dsk/c0t6d0s0
/dev/rdisk/c0t6d0s0
```

and these diskette paths:

```
/dev/diskette
/dev/rdiskette
```

Attempts to mount or access a CD or diskette using these paths will result in an error message.

There are several new commands to help you administer Volume Management on your system as shown in Table 7-4.

Table 7-4 Volume Management Commands

Command	Description
<code>rmmount(1)</code>	Removable media mouter. Used by <code>vold</code> to automatically mount <code>/cdrom</code> and <code>/floppy</code> when media is installed.
<code>volcancel(1)</code>	Cancels a user's request to access a particular CD-ROM or diskette file system

Table 7-4 Volume Management Commands

Command	Description
volcheck(1)	Checks drive for installed media. By default, checks drive pointed to by <code>/dev/diskette</code> .
volmissing(1)	Notifies user when an attempt is made to access media that is no longer in the drive
vold(1)	Volume Management daemon, controlled by <code>/etc/vold.conf</code>

There are also two configuration files to define Volume Management's actions: `/etc/vold.conf` and `/etc/rmmount.conf`. See the `vold.conf(4)` and `rmmount.conf(4)` man pages for descriptions of these files, and see *System Administration Guide, Volume I* for information on managing CD-ROM and floppy devices.

Startup and Shutdown



This chapter describes changes to procedures for booting and shutting down a system.

<i>Booting</i>	<i>page 69</i>
<i>Using the init Command</i>	<i>page 72</i>
<i>Shutting Down</i>	<i>page 74</i>

See *System Administration Guide, Volume I* for detailed descriptions of boot procedures. `man` pages for each command are available on line in the “User Commands” section of *SunOS 4.x Reference Manual*, or in *man Pages(1): User Commands*.

Booting

The Solaris 2.5 boot process makes system administration easier. Some of the major changes include:

- The kernel is self-configuring so you no longer need to rebuild it manually.
- Kernel memory consumption is reduced by automatic loading of devices when first opened.
- File systems are checked only when necessary, improving bootup time.
- The boot block can read UNIX file systems, eliminating boot errors when the boot program moves.
- Third-party bootable devices are supported.

- The SunOS release 4.x `fastboot` command is available only on Solaris 2.5 systems that have the SunOS/BSD Source Compatibility Package installed.
- The SunOS release 4.x `halt` and `reboot` commands have `shutdown(1M)` and `init(1M)` equivalents in SunOS release 5.5.

In the Solaris 2.5 operating environment, the `shutdown` and `init` commands are the preferred way to halt, shut down, or reboot your system. The `reboot` command is available in the Solaris 2.5 operating environment, but it brings the system down quickly, without orderly shutting down services. Table 8-1 shows the SunOS release 5.5 commands that replace those in SunOS release 4.x.

Table 8-1 SunOS Release 5.5 Replacements for `reboot`

SunOS release 4.x	SunOS release 5.5
<code>reboot</code>	<code>shutdown -i 6</code>
<code>fastboot</code>	<code>boot</code>

`boot` *Command Changes*

SunOS release 5.5 has these additional options for the `boot` command:

- Type `boot -r` when you add new hardware or alter its location. This option creates the physical and logical device names, with the logical device name linked to the physical device name.
- Type `boot -v` when you want to see all the system bootup messages; the default is to boot silently. Note the messages are always displayed in the console message buffer.
- Type `boot -a` when you want to be prompted for the name of the system to boot.

Booting From the PROM

Be aware of these changes when booting from PROM:

- The PROM loads `ufsbootblk` from the disk. This file is similar to the previous SunOS release 4.x boot block, except that it is specific to the `ufs` file system. As in SunOS release 4.x, you need to use `installboot(1M)` to put the boot program in the system.
- `/ufsbootblk` opens the boot device and, using the file system you specify, finds and loads `/ufsboot`.

- `/ufsboot` loads the kernel from `/kernel/unix`. SunOS release 4.x systems used `/vmunix`; however, in SunOS release 5.5 the `/kernel` directory contains all kernel modules, including `unix`.
- The kernel, in turn, loads other drivers, such as `sbus`, from the `/kernel/drv` directory. These drivers had to be built as part of the SunOS release 4.x kernel, but can be dynamically loaded in SunOS release 5.5 systems when they are needed.
- The `/sbin/init` command generates processes to set up the system based on the directions in `/etc/inittab`. The next section describes the run levels that `init` uses.

Summary of Boot Differences

Table 8-2 summarizes booting differences.

Table 8-2 Summary of Booting Differences

SunOS release 4.x	SunOS release 5.5	Feature
<code>bootsd</code>	<code>ufsbootblk</code>	Now loads <code>ufsboot</code> from disk
<code>boot program</code>	<code>ufsboot</code>	Now loads <code>unix</code> from disk
<code>/vmunix</code>	<code>/kernel/unix</code>	Bootable kernel image
<code>boot.sun4c.sunos.4.1</code>	<code>inetboot</code>	Mounts and copies <code>unix</code> from network
<code>rc.boot</code> <code>rc.single</code>	<code>/etc/rcS</code>	Mounts <code>/usr</code> and checks file systems
<code>rc.local</code>	<code>/etc/rc2</code> <code>/etc/rc3</code>	System config scripts
<code>/etc/config</code>	<code>modload</code> <code>/etc/system</code>	Customizes system kernel, load modules as needed

Table 8-2 Summary of Booting Differences (Continued)

SunOS release 4.x	SunOS release 5.5	Feature
Prom monitor, single user, multiuser	Run states 0 – 6, and S	System run levels
/dev/sd1g	/dev/dsk/c0t1d0s6	More descriptive logical device names. See “Device Naming Conventions” on page 61.
MAKEDEV	boot -r, add_drv	Makes device nodes

Using the `init` Command

The `init(1M)` command replaces the SunOS release 4.x `fasthalt` command in SunOS release 5.5. Use it to shut down a single-user system. You can use `init` to place the system in a power-down state (`init 0`) or into single-user state (`init 1`).

`init` Command Changes

Note the following changes to the `init` command:

- SunOS release 5.5 system software has eight initialization states (`init` states or run levels). The default `init` state is defined in the `/etc/inittab` file.
- The SunOS release 5.5 `init` command uses a different script for each run level instead of grouping all of the run levels together in the `/etc/rc`, `/etc/rc.boot`, and `/etc/rc.local` files. The files, named by run level, are located in the `/sbin` directory.

SPARC: Installing Solaris Software describes this command in detail. See *man Pages(1): User Commands*.

Changing System Run Levels

The SunOS release 5.5 `init` command lets you control the run level (initialization state) of your system and move easily between various modes of operation. SunOS release 5.5 uses `/sbin/rc` scripts to control each individual run level instead of putting all system states into one file. This enables you to

make changes in a unique file if you create new scripts or modify existing ones. SunOS release 4.x systems controlled run levels, using `/etc/rc`, `/etc/rc.boot` and `/etc/rc.local` files.

SunOS release 4.x had three run levels: prom monitor, single user, and multiuser. These correspond to run levels 0, 1, and 3 in SunOS release 5.5.

Table 8-3 gives an overview of what each run level's `/sbin/rc` script does.

Table 8-3 SunOS Release 5.5 System Initialization Run Levels

Run Level	Default SunOS release 5.5 Function
rc0	Shuts down the system so it is safe to turn off power. Stops system services and daemons. Terminates all running processes. Unmounts all file systems.
rc1	Single-user (system administrator) state for tasks that allow only one user on the system. Stops system services and daemons. Terminates all running processes. Unmounts all file systems.
rc2	Normal multiuser operation without NFS systems exported. Sets the <i>timezone</i> variable. Mounts the <code>/usr</code> file system. Cleans up the <code>/tmp</code> and <code>/var/tmp</code> directories. Loads the network interfaces and starts processes. Starts the <code>cron</code> daemon. Cleans up the <code>uucp tmp</code> files. Starts the <code>lp</code> system. Starts the <code>sendmail</code> daemon.
rc3	Normal multiuser operation of a file server with NFS systems exported. Completes all of the tasks in run level 2. Starts the NFS system daemons.
rc4	Alternative multiuser state (not used).

Table 8-3 SunOS Release 5.5 System Initialization Run Levels (Continued)

Run Level	Default SunOS release 5.5 Function
rc5	Software reboot, prompts for a boot device other than the EEPROM default.
rc6	Reboot. Kills all active processes. Unmounts the file systems and reboots to the <i>initdefault</i> entries in <i>/etc/inittab</i> .
rcS,s	Single-user state, running with some file systems mounted and accessible.

Shutting Down

Use the `shutdown(1M)` command when shutting down a system with multiple users. The command sends a warning to all logged-in users, and, after 60 seconds, shuts the system down to single-user state.

- The SunOS release 4.x `fasthalt` commands are available only on SunOS release 5.5 systems that have the SunOS /BSD Source Compatibility Package installed.
- The SunOS release 4.x `halt` and `reboot` commands have `shutdown` and `init` equivalents.

See *System Administration Guide, Volume I* for detailed descriptions of shutdown procedures. `man` pages for each command are available in *man Pages(1): User Commands*.

In SunOS release 5.5, the `shutdown` command is the preferred way to halt or shut down a system. `shutdown` and `init` use `rc` scripts to kill running processes. The `halt` command is available in SunOS release 5.5, but it stops the system quickly, without orderly shutting down services. Table 8-4 shows the SunOS release 5.5 commands that replace those in the SunOS release 4.x system.

Table 8-4 SunOS Release 5.5 Replacements for `shutdown`

SunOS release 4.x	SunOS release 5.5
<code>halt</code>	<code>init 0</code>
<code>fasthalt</code>	<code>init 0</code>

The `shutdown` and `init` commands accept a numerical “run-level” argument that controls the shutdown sequence. See the `shutdown` and `init` man pages of *man Pages(1M): System Administration Commands* for information about the run-level numbers.

Changes to the shutdown Command

The SunOS release 5.5 `shutdown` command includes only the options in Table 8-5. This command and its options are described in *System Administration Guide, Volume I*.

Table 8-5 SunOS Release 5.5 `shutdown` Command Options

Option	SunOS release 5.5 Function
-g	Selects “grace” period before <code>shutdown</code> begins.
-i [<i>init state</i>]	Specifies an initial run level (see Table 8-3 on page 73).
-y	Runs <code>shutdown</code> without asking confirmation questions. Assumes a “yes” response to all questions.

By default, the SunOS release 5.5 `shutdown` command asks you to confirm before an actual `shutdown` begins, but it has an option (-y) that allows it to run without operator intervention.

The SunOS release 5.5 `shutdown` command does not support any of the options available with SunOS release 4.x `shutdown`. Therefore, SunOS release 4.x options are available only in BSD source compatibility mode on Solaris 2.5 systems.

See Appendix A, “Commands Reference Table,” for a summary of changes. See the `shutdown` man page for information about how the command works.

Using the fasthalt and fastboot Commands

The SunOS release 4.x `fastboot` and `fasthalt` commands are available if you are running the SunOS/BSD Source Compatibility Package on Solaris 2.5 systems. The file-system checking features of these commands are not appropriate to Solaris 2.5.

Using the halt and reboot Commands

The `halt` and `reboot` commands do not run the `rc` scripts in `/sbin`, so they are not recommended. Since the `halt` and `reboot` commands in SunOS release 5.5 systems are not available on other AT&T SVR4 systems, both commands have `shutdown` and `init` equivalents.

File-System Administration

This chapter familiarizes you with changes to file system layout and the changes to file systems, virtual file systems, directories, and files. The chapter also describes changes to file system administration including:

- Mounting file systems
- Monitoring file systems
- Sharing file systems
- Creating new file systems
- Checking file systems
- Backing up and restoring files

This chapter contains the following sections:

<i>File-System Changes</i>	<i>page 78</i>
<i>Default File Systems and Directories</i>	<i>page 79</i>
<i>Virtual File-System Architecture</i>	<i>page 82</i>
<i>Directory and File Changes</i>	<i>page 87</i>
<i>Using File System Administration Commands</i>	<i>page 95</i>

For more information on understanding and managing file systems, see *System Administration Guide, Volume I*.

File-System Changes

SunOS release 5.5 and SunOS release 4.x file systems are similar, but there are changes in the locations and names of system directories and files; there are also new file systems and new pseudo file systems, and one directory was removed.

Some of the changes to file system locations and names are:

- The `/dev` directory has changed from a flat directory to a hierarchical one.
- The `/etc` directory has changed and contains specific system configuration information. Several files and subdirectories have been added, removed, or changed in SunOS release 5.5.
- The `/etc/vfstab` tab file replaces `/etc/fstab`.
- The `/etc/lp` directory replaces `/etc/printcap`.
- The SunOS release 5.5 `/sbin` directory contains the `rc` scripts used to alter system run levels as well as the `rcs` script used to initialize the system prior to mounting file systems.
- The SunOS release 5.5 `/usr` directory contains sharable files and executables provided by the system.
- The `/var` directory contains files that change size during normal operation. Several files and subdirectories in the `/var` directory have been added, removed, or changed in SunOS release 5.5.
- The `/var/mail` directory replaces `/var/spool/mail`.
- The `/sys` directory is no longer needed because the kernel is dynamically loaded.
- The `/RFS` file system has been removed.
- The `terminfo` database replaces `termcap`.

Pseudo File Systems

The `TFS` pseudo file system is not included in the SunOS release 5.5.

The added pseudo file systems in SunOS release 5.5 are:

- The `CACHEFS` pseudo file system can be used to improve performance of slow devices such as CD-ROM.

- The `PROCFS` pseudo file system resides in memory and contains a list of active processes, by process number, in the `/proc` directory. See the `proc(4)` manual page.
- The `FDFS` pseudo file system provides explicit names for opening files using file descriptors.
- The `FIFOFS` pseudo file system contains pipe files that give processes common access to data.
- The `NAMEFS` pseudo file system is used mostly by `STREAMS` for dynamic mounts of file descriptors on top of files.
- The `SWAPFS` pseudo file system is the default swap device when the system boots or you create additional swap space.

Added File Systems

The following file systems are included in the SunOS release 5.5 directory structure:

- The kernel, now called `unix`, and the kernel modules, are stored in the `/kernel` directory.
- The optional `/opt` file system can be used to store third-party or unbundled software. If `/opt` is not a separate file system, it may be a symbolic link to `/usr/opt`.
- The `/vol` file system provides the default file system for the volume management daemon, `vold(1M)`. See the `volfs(7)` man page.

Default File Systems and Directories

The SunOS release 5.5 file system is hierarchical. Figure 9-1 graphically depicts SunOS release 5.5 default directories and file systems (indicated by dotted lines).

Subdirectories shown are just a sample of what the directory or file system actually holds. Figure 9-1 gives a brief description of each.

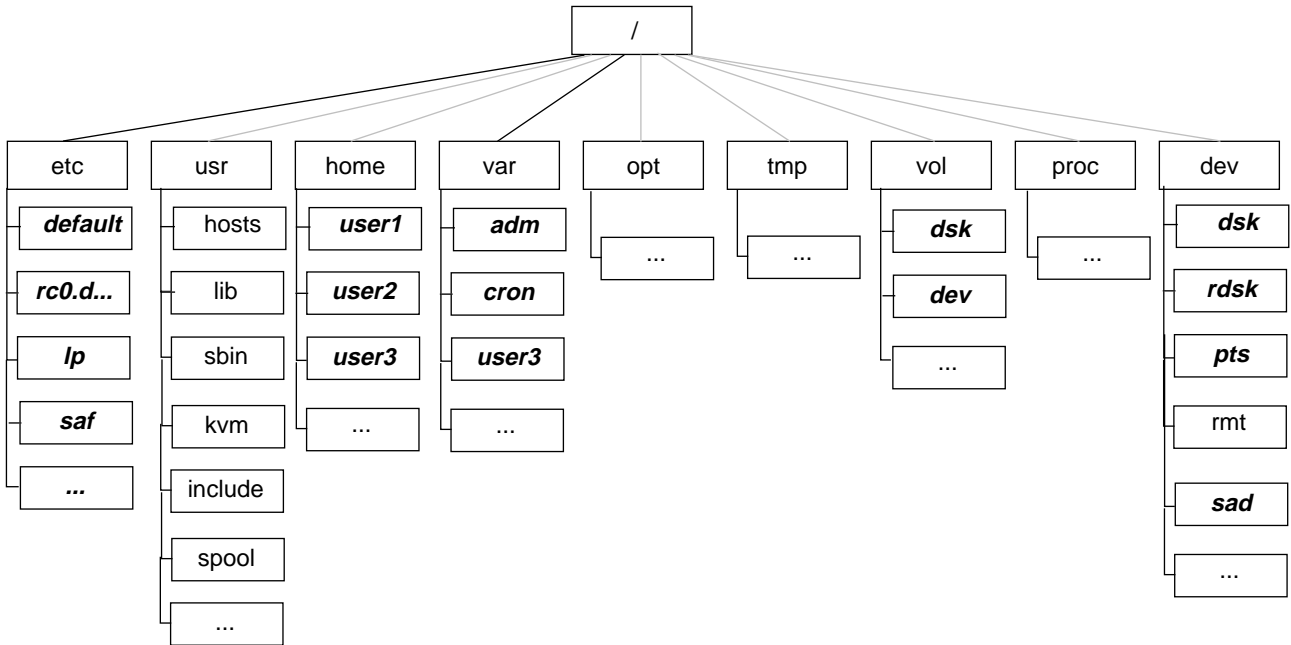


Figure 9-1 Solaris Release 2.4 Default File Systems and Directory Hierarchy

Solaris 2.5 software contains a default set of file systems and directories, and uses a set of conventions to group similar types of files together. Table 9-1 lists SunOS release 5.5 default file systems and directories with a brief description.

Table 9-1 Default Solaris 2.4 File Systems and Directories

File System or Directory	Type	Description
/	File system	The top of the hierarchical file tree. The root directory contains the directories and files critical for system operation, such as the kernel (<code>/kernel/unix</code>), the device drivers, and the programs used to boot the system. It also contains the mount point directories where local and remote file systems can be attached to the file tree.
/etc	Directory	Contains system files used in system administration
/usr	File system	Contains architecture-dependent and -independent sharable files. Files such as man pages that can be used on all types of systems are in <code>/usr/share</code> .
/home	File system	The mount point for the users' home directories, which store users' work files. By default, <code>/home</code> is now an automounted file system.
/var	Directory	Contains system files and directories that are likely to change or grow over the life of the local system. These include system logs, <code>vi</code> and <code>ex</code> backup files, and <code>uucp</code> files.
/opt	File system	Mount point for optional third-party software. On some systems <code>/opt</code> may be a <code>ufs</code> file system on a local disk partition.
/tmp	File system	Temporary files, cleared each time the system is booted or <code>/tmp</code> is unmounted
/vol	File system	Contains directories for removable media, managed by <code>vold(1M)</code>
/proc	File system	Contains a list of active system processes, by number. Does not use any disk space
/sbin	Directory	Essential executables used in the booting process and in manual system recovery

Virtual File-System Architecture

SunOS release 5.5 features a virtual file system (VFS) architecture that simplifies file system management for systems that support multiple file systems.

Over the years, several different UNIX file systems were developed, each with its own set of commands for file system management. Learning all the variations can be confusing and difficult. SunOS release 5.5 addresses this issue with a set of generic commands for file system management. These commands comprise a common VFS interface that makes differences between file systems transparent with respect to maintenance. The subsections below list a summary of supported file systems and the generic file system commands.

Supported File-System Types

Most file system types included in SunOS release 4.x are also included in SunOS release 5.5. There is one exception: The translucent file system (TFS) type has been withdrawn from SunOS release 5.5. Table 9-2 summarizes file-system type availability in SunOS release 4.x and SunOS release 5.5.

Table 9-2 Summary of File System Types

Category	Name	Description	SunOS Release 4.x	SunOS Release 5.5
Disk-based	UFS	UNIX file system	Yes	Yes
	HSFS	CD-ROM file system	Yes	Yes
	PCFS	PC file system	Yes	Yes
Network-based	NFS	Sun's distributed computing file system	Yes	Yes
Pseudo	SPECFS	Device Special file system	Yes	Yes
	TMPFS	/tmp Temporary file system	Yes	Yes
	LOFS	Loopback file system	Yes	Yes
	TFS	Translucent file system	Yes	No
	PROCFS	Process Access file system	No	Yes
	FDFS	File Descriptor file system	No	Yes

Table 9-2 Summary of File System Types (Continued)

Category	Name	Description	SunOS Release 4.x	SunOS Release 5.5
	FIFOFS	FIFO/Pipe file system	No	Yes
	NAMEFS	Name file system	No	Yes
	SWAPFS	Swap file system	No	Yes
	CACHEFS	Cache file system	No	Yes

For more information on file systems, see the `proc(4)` and `fd(4)` man pages and *System Administration Guide, Volume I*.

The Cache File System (CACHEFS)

The Cache File System can be used to improve performance of remote file systems or slow devices such as CD-ROM. When a file system is cached, the data read from the remote file system or CD-ROM is stored in a cache on the local system.

Swap File Changes

SWAPFS is SunOS release 5.5's default swap device when the system boots or you create additional swap space. This swap device uses physical memory as swap space, but also requires physical swap space on disk.

In SunOS release 4.x systems, the default physical swap device depends on the system configuration. Standalone systems default `sd0b` and diskless systems get their swap files from the `bootparam` server. SunOS release 5.5 uses the swap file as the default dump device instead of specifying a file on disk.

Unsupported SVR4 File System Types

Table 9-3 shows SVR4 file system types that are not supported in SunOS release 5.5.

Table 9-3 SVR4 File System Types Not Supported

Name	Description
BFS	Boot file system
S5	System V file system
xnamefs	XENIX semaphore file system

Generic File-System Commands

Most file system administration commands have a *generic* and a *file system* component. Use the generic commands, which call the file system component. Table 9-4 lists the generic file system administrative commands, which are located in the `/usr/bin` directory.

Table 9-4 Generic File System Administrative Commands

Command	Description
<code>clri(1M)</code>	Clears inodes
<code>df(1M)</code>	Reports the number of free disk blocks and files
<code>ff(1M)</code>	Lists file names and statistics for a file system
<code>fsck(1M)</code>	Checks the integrity of a file system and repairs any damage found
<code>fsdb(1M)</code>	File system debugger
<code>fstyp(1M)</code>	Determines the file-system type
<code>labelit(1M)</code>	Lists or provides labels for file systems when copied to tape (for use by the <code>volcopy</code> command only)
<code>mkfs(1M)</code>	Makes a new file system
<code>mount(1M)</code>	Mounts file systems and remote resources
<code>mountall(1M)</code>	Mounts all file systems specified in a file-system table
<code>ncheck(1M)</code>	Generates a list of path names with their i-numbers
<code>umount(1M)</code>	Unmounts file systems and remote resources
<code>umountall(1M)</code>	Unmounts all file systems specified in a file-system table

Table 9-4 Generic File System Administrative Commands (Continued)

Command	Description
<code>volcopy(1M)</code>	Makes an image copy of a file system

Most of these commands also have a file system counterpart.



Caution – Do not use the file system commands directly. If you specify an operation on a file system that does not support it, the generic command displays this error message: *command*: Operation not applicable for FSType *type*.

Syntax of Generic Commands

Most of these commands use this syntax:

```
command [-F type] [-V] [generic-options] [-o specific-options] [special|mount-point]
[operands]
```

The options and arguments to the generic commands are:

-F *type*

Specifies the type of file system. If you do not use this option, the command looks for an entry that matches *special* or *mount point* in the `/etc/vfstab` file. Otherwise, the default is taken from the file `/etc/default/fs` for local file systems and from the file `/etc/dfs/fstypes` for remote file systems.

-V

Echoes the completed command line. The echoed line may include additional information derived from `/etc/vfstab`. Use this option to verify and validate the command line. The command is not run.

generic-options

Options common to different types of file systems.

-o *specific-options*

A list of options specific to the type of file system. The list must have the following format: `-o` followed by a space, followed by a series of *keyword* [=value] pairs separated by commas with no intervening spaces.

special | *mount-point*

Identifies the file system. The name must be either the mount point or the special device file for the slice holding the file system. For some commands, the *special* file must be the raw (character) device and for other commands it must be the block device. In some cases, this argument is used as a key to search the file `/etc/vfstab` for a matching entry from which to obtain other information. In most cases, this argument is required and must come immediately after *specific-options*. However, it is not required when you want a command to act on all the file systems (optionally limited by type) listed in the `/etc/vfstab` file.

operands

Arguments specific to a type of file system. See the specific man page of the command (for example, `mkfs_ufs`) for a detailed description.

System-wide Default File-System Type

The default remote file system type is `/etc/dfs/fstype`. The default local file system type is `/etc/default/fs`. See the `default_fs(4)` man page for more information.

Command Locations

In previous SunOS releases, all file-system commands were located in the `/etc` directory. In SunOS release 5.5, file-system commands are organized into separate hierarchies for convenience. All of the file-system commands are included in `/usr/lib/fs/fstype`. Commands needed before `/usr` is mounted are duplicated in `/etc/fs/fstype`.

All of the generic commands are located in `/usr/sbin`. The commands needed before `/usr` is mounted are duplicated in `/sbin`.

Table 9-5 lists the locations of the file system commands.

Table 9-5 Locations of File System Commands

Type	Location of Primary Version	Location of Duplicate Version (root)
Generic	<code>/usr/sbin</code>	<code>/sbin</code>
Specific	<code>/usr/lib/fs</code>	<code>/etc/fs</code>

Directory and File Changes

This section describes the changes to directories and files between SunOS release 4.x and SunOS release 5.5.

/dev Directory

The `/dev` directory has changed from a flat directory to a hierarchical one. Table 9-6 describes the subdirectories that have been added.

Table 9-6 Additions to the `/dev` Directory

Subdirectory	Description
<code>/dev/dsk</code>	Contains block disk devices
<code>/dev/rdsk</code>	Contains raw disk devices
<code>/dev/pts</code>	Contains pseudo terminal (<code>pty</code>) slave devices
<code>/dev/rmt</code>	Contains raw tape devices
<code>/dev/sad</code>	Contains entry points for the STREAMS Administrative Driver
<code>/dev/term</code>	Contains terminal devices

/etc Directory

The `/etc` directory contains system configuration information. Several files and subdirectories in the SunOS release 4.x `/etc` directory have been added, removed, or changed in SunOS release 5.5.

- File system commands, such as `mount *`, have been moved to subdirectories of the `/usr/lib/fs` directory.
- The SunOS release 4.x `/etc/fstab` file has been replaced by `/etc/vfstab`.

Initialization scripts, such as `rc`, `rc.boot`, `rc.local`, and `rc.single`, are not available in SunOS release 5.5. They are replaced by the scripts shown in Table 9-7 which are run by their corresponding run control files.

Table 9-7 Initialization Scripts and Their Run Control Files

Scripts	Run Control Files
<code>/etc/rc0.d</code>	<code>/sbin/rc0</code>
<code>/etc/rc1.d</code>	<code>/sbin/rc1</code>
<code>/etc/rc2.d</code>	<code>/sbin/rc2</code>
<code>/etc/rc3.d</code>	<code>/sbin/rc3</code>
<code>/etc/rc4.d</code>	<code>/sbin/rc4</code>
<code>/etc/rc5.d</code>	<code>/sbin/rc5</code>
<code>/etc/rc6.d</code>	<code>/sbin/rc6</code>
<code>/etc/rcS.d</code>	<code>/sbin/rcS</code>

Table 9-8 describes the subdirectories that have been added to the `/etc` directory for SunOS release 5.5.

Table 9-8 Additions to the /etc Directory

Subdirectory	Description
<code>/etc/default</code>	Defines default system configuration
<code>/etc/inet</code>	Defines Internet services configuration
<code>/etc/lp</code>	Defines LP system configuration
<code>/etc/opt</code>	Defines installed optional software
<code>/etc/rcn.d</code>	Defines run-state transition operations
<code>/etc/saf</code>	Defines Service Access Facility (SAF) configuration

The /etc/vfstab File

In SunOS release 5.5, the virtual file system file `/etc/vfstab` replaces the `/etc/fstab` file. In the virtual file system architecture, the `/etc/vfstab` file provides default file system parameters used by the generic commands for file system management. For information about these commands, see “Generic File-System Commands” on page 84.

In addition to the name change, the `/etc/vfstab` file is different from the `/etc/fstab` file in the following ways:

- A `device to fsck` field has been added to specify the names of raw devices to be checked by `fsck`.
- An `automount` field has been added to control the routine mounting of file system by `mountall` (the automount daemon does not use this field).
- The `freq` field, which specified the number of days between dumps, has been eliminated.

The file-system table has seven fields, each separated by a tab. Table 9-9 explains the field entries.

Note – You must have an entry in each field in the `/etc/vfstab` file. If there is no value for a field, be sure to type a dash (-).

Table 9-9 `/etc/vfstab` File Field Names and Content

Field Name	Content
device to mount	<p>The entry in this field may be any of the following:</p> <p>The block special device for local <code>ufs</code> file systems (for example, <code>/dev/dsk/c0t0d0s0</code>)</p> <p>The resource name for remote file systems (for example, <code>myserver:/export/home</code> for an <code>nfs</code> system)</p> <p>The name of the slice on which to swap (for example, <code>/dev/dsk/c0t3d0s1</code>)</p> <p>The <code>/proc</code> directory and <code>proc</code> file system type</p> <p>CD-ROM as <code>hfs</code> file system type</p> <p><code>/dev/diskette</code> as <code>pcfs</code> or <code>ufs</code> file system type</p> <p>This field is also used to specify swap file systems. For more information on remote file systems, see <i>NFS Administration Guide</i>.</p>
device to fsck	<p>The raw (character) special device that corresponds to the file system identified by the <code>device to mount</code> field (for example, <code>/dev/rdisk/c0t0d0s0</code>). This field determines the raw interface that is used by <code>fsck</code>. Use a dash (-) when there is no applicable device, such as for a read-only file system or a network-based file system.</p>
mount point	<p>The default mount point directory (for example, <code>/usr</code> for <code>/dev/dsk/c0t0d0s6</code>).</p>
FS type	<p>The type of file system identified by the <code>device to mount</code> field.</p>

Table 9-9 /etc/vfstab File Field Names and Content (Continued)

Field Name	Content
<code>fsck pass</code>	The pass number used by <code>fsck</code> to determine whether to check a file system. When the field contains a dash (-), the file system is not checked. When the field contains a value of 1 or more, the file system is checked; non- <code>ufs</code> file systems with a 0 <code>fsck pass</code> are checked. For <code>ufs</code> file systems only, when the field contains a 0, the file system is not checked. When <code>fsck</code> is run on multiple <code>ufs</code> file systems that have <code>fsck pass</code> values greater than 1 and the <code>preen</code> option (<code>-o p</code>) is used, <code>fsck</code> automatically checks the file systems on different disks in parallel to maximize efficiency. When the field contains a value of 1, the file system is checked sequentially. Otherwise, the value of the pass number does not have any effect. In SunOS 5.3 system software, the <code>fsck pass</code> field does <i>not</i> explicitly specify the order in which file systems are checked.
<code>automount?</code>	yes or no for whether the file system should be automatically mounted by <code>mountall</code> when the system is booted. An <code>auto</code> in the fourth column of your SunOS release 4.x <code>/etc/fstab</code> would translate to a “yes” in this column; a <code>noauto</code> , a “no.” Note that this field has nothing to do with the <code>automount</code> program.
<code>mount options</code>	A list of comma-separated options (with no spaces) that are used in mounting the file system. Use a dash (-) to show no options. See the <code>mount(1M)</code> man page for a list of the available options.

Figure 9-2 is an example of a SunOS release 5.5 `/etc/vfstab` file.

#device #to mount #	device to fsck	mount point	FS type	fsck pass	auto mount?	mount options
/dev/dsk/c0t0d0s0	/dev/rdisk/c0t0d0s0	/	ufs	1	no	-
/proc	-	/proc	proc	-	no	-
swap	-	/tmp	tmpfs	-	yes	-
/dev/dsk/c0t0d0s6	/dev/rdisk/c0t0d0s6	/usr	ufs	2	no	-
/dev/dsk/c0t3d0s7	/dev/rdisk/c0t3d0s7	/files7	ufs	3	no	-
oak:/export/home1/ignatz	-	/home/ignatz	nfs	-	yes	rw,intr

Figure 9-2 SunOS Release 5.5 `/etc/vfstab` file

For detailed information about the `/etc/vfstab` file, see *System Administration Guide, Volume I*.

The /etc/shadow File

The `/etc/shadow` file, new to SunOS release 5.5, includes entries that force password aging for individual user login accounts. The `/etc/shadow` file also contains encrypted passwords. The `/etc/shadow` file does not have general read permissions. This prevents general access to the encrypted passwords that formerly appeared in the `/etc/passwd` file.

/sbin Directory

The SunOS release 5.5 `/sbin` directory contains the `rc` scripts used to alter system run levels as well as the `rcs` script used to initialize the system prior to mounting file systems. See the `rc` man pages in *man Pages(1M): System Administration Commands* and “Changing System Run Levels” on page 72 for a description of the scripts.

`/usr` Directory

The SunOS release 5.5 `/usr` directory contains sharable files and executables provided by the system. Table 9-10 describes the subdirectories that have been added to the SunOS release 4.x `/usr` directory for SunOS release 5.5.

Table 9-10 Additions to the `/usr` Directory

Subdirectory	Description
<code>/usr/ccs</code>	C compilation systems
<code>/usr/snadm</code>	Executables and other files used by <code>admintool</code>

Table 9-11 shows files that have been moved from the SunOS release 4.x `/usr` directory in SunOS release 5.5.

Table 9-11 Files Changed in the `/usr` Directory

SunOS Release 4.x Location	SunOS Release 5.5 Location
<code>/usr/5bin</code>	<code>/usr/bin</code>
<code>/usr/5include</code>	<code>/usr/include</code>
<code>/usr/5lib</code>	<code>/usr/lib</code>
<code>/usr/etc</code>	<code>/usr/sbin</code>
<code>/usr/old</code>	Contents removed
<code>/usr/xpg2bin</code>	<code>/usr/bin</code>
<code>/usr/xpg2lib</code>	<code>/usr/lib</code>
<code>/usr/xpg2include</code>	<code>/usr/include</code>

Appendix E, “`/` and `/usr` File Systems Changes,” contains tables with detailed information about the directories and files in each of these file systems.

`/var` Directory

The `/var` directory contains files that change sizes during normal operation. Several files and subdirectories in the `/var` directory have been added, removed, or changed for SunOS release 5.5.

- The `/var/opt/packagename` directory contains software package objects that change sizes, such as `log` and `spool` files.

- The `/var/sadm` directory contains databases maintained by the software package management utilities.
- The `/var/saf` directory contains Service Access Facility (SAF) logging and accounting files.
- The SunOS release 4.x `/var/spool/mail` directory has been moved to `/var/mail` in SunOS release 5.5.

Two directories were added to SunOS release 5.x file system: `/kernel` and `/opt`.

`/kernel` Directory

The SunOS release 5.5 `/kernel` directory contains the operating system kernel and kernel-level object modules, which were in `/sys` in SunOS release 4.x. Table 9-12 describes the subdirectories that have been added to the `/kernel` directory for the SunOS release 5.5.

Table 9-12 Additions to the `/kernel` Directory

Subdirectory	Description
<code>/kernel/drv</code>	Device driver and pseudo-device driver modules
<code>/kernel/exec</code>	Kernel modules to run ELF or a.out executable files
<code>/kernel/fs</code>	Kernel modules that implement file systems such as <code>ufs</code> , <code>nfs</code> , <code>proc</code> , <code>fifo</code> , etc.
<code>/kernel/misc</code>	Miscellaneous modules
<code>/kernel/sched</code>	Modules containing scheduling classes and corresponding dispatch tables
<code>/kernel/strmod</code>	STREAMS modules
<code>/kernel/sys</code>	Loadable system calls such as system accounting and semaphore operations
<code>/kernel/unix</code>	Operating system kernel, loaded at boot time

`/opt` Directory

The SunOS release 5.5 `/opt` directory contains optional add-on application software packages. These packages were installed in `/usr` on SunOS release 4.x.

/sys Directory

The `/sys` directory has been retired. Its files, used to reconfigure the kernel, have been made obsolete by the dynamic kernel.

Using File System Administration Commands

The file system administration commands that have changed from SunOS release 4.x to SunOS release 5.5 include those for:

- Mounting file systems
- Monitoring file systems
- Sharing file systems
- Creating a new file system
- Checking a file system
- Backing up and restoring files

When you are ready to administer file systems on your SunOS release 5.5, see *System Administration Guide, Volume I* for details on performing the tasks involved.

Mounting File Systems and `autofs`

The biggest change to the mounting capability is automatic mounting or `autofs`. The `autofs` program automatically mounts directories when you access them using, for example `cd(1)` or `ls(1)`. This capability includes file hierarchies, CD-ROM, and diskette file systems.

`autofs` starts automatically when the system enters run level 3, or you can invoke it from a shell command line.

The `autofs` works with the file systems specified in *maps*. These maps can be maintained as NIS, NIS+, or local files. The `autofs` maps can specify several remote locations for a particular file. This way, if one of the servers is down, `autofs` can try to mount from another system. You can specify which servers are preferred for each resource in the maps by assigning each server a weighting factor.

Mounting some file hierarchies with `autofs` does not exclude the ability to mount others with the `mount` command. A diskless system *must* have entries for `/` (root), `/usr`, and `/usr/kvm` in the `/etc/vfstab` file. Because shared file systems should always remain available, do not use `autofs` to mount `/usr/share`.

The following example shows how to manually mount a file system listed in the `/etc/vfstab` file using the `mount` command.

1. **Change to the directory in which you want to create the mount point.**
2. **Create the mount-point directory.**
3. **Specify either the mount point or the block device.**
It is usually easier to specify the mount point. The rest of the information is read from `/etc/vfstab`.
4. **Become root and give the `mount` command, specifying either the mount point or the block device.**
It is usually easier to specify the mount point. The rest of the information is read from `/etc/vfstab`.

```
# mount mount-point
```

The file system is now mounted.

For instructions showing how to mount different types of file systems using `mount` with or without options, see *System Administration Guide, Volume I*.

Changes to the `mount` Command

Some of the names and forms of the SunOS release 5.5 `mount` commands are different from those in SunOS release 4.x as shown in Table 9-13.

Table 9-13 SunOS Release 4.x and SunOS Release 5.5 `mount` Command Differences

SunOS Release 4.x	SunOS Release 5.5
<code>mount</code>	<code>mount</code>
<code>mount -a</code>	<code>mountall</code>
<code>umount</code>	<code>umount</code>

Table 9-13 SunOS Release 4.x and SunOS Release 5.5 mount Command Differences

SunOS Release 4.x	SunOS Release 5.5
<code>umount -a</code>	<code>umountall</code>
<code>exportfs</code>	<code>share</code>
<code>exportfs -u</code>	<code>unshare</code>
<code>showmount -a</code>	<code>dfmounts</code>
<code>showmount -e</code>	<code>dfshares</code>

See Appendix A, “Commands Reference Table,” for more information on changes to these commands.

Automatic Mounting of /cdrom and /floppy

In this release, the CD-ROM and diskette file systems are automatically mounted in `/cdrom` and `/floppy` when removable media is inserted into these drives. Since these file systems are now managed by the Volume Management daemon, `vold(1M)`, you cannot mount these devices yourself. See “Using Volume Management” on page 66 for more information.

Specifying File Systems in the /etc/vfstab File

In the SunOS release 5.5 system, you need to list file systems that you want mounted at system startup in your `/etc/vfstab`, instead of in the SunOS release 4.x `/etc/fstab` file. The format of `/etc/vfstab` differs from that of `/etc/fstab`. For a discussion of the `/etc/vfstab` file, see “The `/etc/vfstab` File” on page 89.

Monitoring File Systems

Table 9-14 shows the file and directory monitoring commands and changes, where they apply.

Table 9-14 File and Directory Monitoring Commands

Command	Information Provided	Change (if applicable)
ls	Size, age, permissions, owner of files	None
du	Total size of directories and their contents	None
df	Disk space occupied by file systems, directories, or mounted resources; used and available disk space	The SunOS release 4.x version of this command provides a different output format containing somewhat different output than the SunOS release 5.5 <code>df</code> command. The SunOS release 5.5 <code>-k</code> option provides output formats similar to those in the SunOS release 4.x command. The SunOS release 4.x <code>df -t filesystem</code> type reports on files of the specified type, whereas the SunOS release 5.5 <code>df -t</code> command prints full listings with totals.
quot	Number of blocks owned by users	None
find	Names of files meeting search criteria	The following SunOS release 4.x option is not available in the SunOS release 5.5 command: <code>-n cpio-device</code> Write the current file on device in <code>cpio -c</code> format.

Sharing File Systems

File systems were “exported” in SunOS release 4.x to make them available to other systems. This was done through the `/etc/exports` file and the `exportfs` command. However, only NFS systems could be exported.

In SunOS release 5.5, this same concept is referred to as “sharing resources,” and it has been expanded to include more file systems. File systems are shared with the `share(1M)` and `shareall(1M)` commands. The `share` command is similar to the `exportfs pathname` command, while `shareall` is similar to the `exportfs -a` command.

The `share -F fstype` option specifies the type of file system to be shared. If the `-F` option is not specified, `share` uses the first file-system type listed in the `/etc/dfs/dfstab` file.

File systems that you want to be shared automatically should have `share` command entries in the `/etc/dfs/dfstab` file (which replaces the `/etc/export` file). The commands specified in this file are run automatically when the system enters run level 3 (multiuser mode with network file sharing).

Example of `/etc/dfs/dfstab` file entries

The following entry gives clients on `mercury`, `venus`, and `mars` read-write access to `/export/home1`; the second entry gives clients on `saturn` and `jupiter` read-only access to `/export/news`.

```
share -F nfs -o rw=mercury:venus:mars -d "Home Dir" /export/home1
share -F nfs -o ro=saturn:jupiter -d "News Postings" /export/news
```

When the system is running in multiuser mode, these file systems are available to the clients listed. The `share` command displays all resources shared by the local system:

```
% share
-           /export/home1  rw=mercury:venus:mars  "Home Dir"
-           /export/news   ro=saturn:jupiter     "News Postings"
```

Creating New File Systems

You define, specify, and create a new file system using either the `newfs(1M)` or the `mkfs(1M)` command. The following sections highlight changes in the `newfs` and `mkfs` commands.

The `newfs` Command

The SunOS release 5.5 `newfs` command is a convenient front end to the `mkfs` command. The `newfs` command does not support the virtual file system architecture; it is intended for creating `ufs`-type file systems only. When you use `newfs`, it calls and passes arguments to `mkfs`, which does the real work when creating a `ufs` file system.

The `newfs` command accepts only names that conform to the SunOS release 5.5 device naming conventions (see “Device Naming Conventions” on page 61).

The `mkfs` Command

The SunOS release 5.5 `mkfs` command differs significantly from the SunOS release 4.x version of the command. The SunOS release 5.5 version provides for different file-system types, and its command syntax is entirely different (see “Generic File-System Commands” on page 84). Like `newfs`, `mkfs` accepts only names conforming to the SunOS release 5.5 device naming conventions.

Although `mkfs` now supports different types of file systems, in practice it is almost always used to create `ufs` file systems. However, `mkfs` isn’t usually run directly; it is usually called by the `newfs` command.

See *man Pages(1): User Commands* for additional details.

Checking File Systems

The SunOS release 5.5 `fsck(1M)` command differs significantly from the SunOS release 4.x version of the command. In keeping with the virtual file-system (VFS) architecture, the `fsck` file-checking utility has two parts:

- A generic command that is called first, regardless of the type of file system.

- A specific command that is called by the generic command, depending on the type of the target file system (see “Generic File-System Commands” on page 84).

In addition, `fsck` accepts only names conforming to the SunOS release 5.5 device naming conventions. For more information about the SunOS release 5.5 device naming conventions, see “Device Naming Conventions” on page 61.

The `fsck` command performs faster consistency checks at mount time. In addition, the SunOS release 5.5 software does not require you to reboot the system after running `fsck` on the root and `/usr` file systems. This results in faster system start up compared to previous SunOS releases. The `fsck -m` command enables you to skip checking for file systems that are clean. See *man Pages(1): User Commands* for additional details.

Backing Up and Restoring Files

This section discusses the changes to backup and restore commands between SunOS release 4.x and SunOS release 5.5 and describes how to use `ufsdump`, `ufsrestore`, `dd`, `tar`, and `cpio` commands.

SunOS release 4.x supported several utilities for backing up and restoring files: `dump`, `restore`, `tar`, `cpio`, `dd`, and `bar`, as well as the unbundled Backup CoPilot program. This release supports all of these utilities except `bar` and Backup CoPilot. SunOS release 4.x `bar` files can be restored on a SunOS release 5.5 system, but you cannot create new `bar` files. The `dump(8)` and `restore(8)` commands were renamed `ufsdump(1M)` and `ufsrestore(1M)`. Files created with the SunOS release 4.x `dump` command can be restored on a SunOS release 5.5 system with `ufsrestore`.

SunOS release 5.5 has two additional utilities for copying file systems: `volcopy(1M)` and `labelit(1M)`.

The `ufsdump` Command

The `ufsdump` command accepts the same command syntax as the SunOS release 4.x `dump` command. `ufsdump` also accepts options listed in Table 9-15.

Table 9-15 `ufsdump` Command Options Not Available With the `dump` Command

Option	Function
-l	Autoload. When reaching the end of a tape (before completing the dump), take the drive off line and wait up to two minutes for the tape drive to be ready again. This gives autoloading (stackloader) tape drives a chance to load a new tape. If the drive is ready within two minutes, continue. If it is not ready after two minutes, prompt an operator to load another tape, as usual, and wait.
-o	Off line. When finished with a tape or diskette (completing the dump or reaching the end of the medium), take the drive off line. In the case of a diskette drive, also eject the diskette. In the case of a tape drive, also rewind the tape. This prevents another process that rushes in to use the drive from inadvertently converting the data.
-s	Estimate size of dump. Determine the amount of space that is needed to perform the dump, without actually doing it, and output a single number indicating the estimated size of the dump in bytes. This is most useful for incremental backups.

Unlike `dump`, `ufsdump` can detect the end of medium, so it is no longer necessary to use the `-s` size option to force dump programs to move to the next tape before reaching the end. Nevertheless, to ensure compatibility with older versions of the `restore` command, the `-s` option has been retained in `ufsdump`.

Even though `ufsdump` now can detect the end of medium, it has no way to predict the number of diskettes or tapes needed for a dump—unless you specify the medium size with the `-s` option. Therefore, the messages displayed at the start of a backup do not indicate the number of diskettes or tapes required, unless you have specified the medium size.

The `-w` and `-W` options behave a little differently in SunOS release 5.5. In SunOS release 4.x, these options list all file systems that are scheduled for backup according to the backup frequencies specified in the `/etc/fstab` file. Since the SunOS release 5.5 equivalent file, `/etc/vfstab`, has no provision for

specifying backup frequencies, these options now assume that each file system will be backed up daily. Therefore, they now list any file systems that have not been backed up within a day.

When performing backups across the network (backing up local file systems to a remote tape drive), use the device naming convention that's appropriate for the system with the tape drive. If the system with the tape drive is a SunOS release 5.5 system, use the device naming convention to identify the tape drive; otherwise, use the SunOS release 4.x convention.

The `ufsrestore` Command

The `ufsrestore` command in SunOS release 5.5 is similar to the `restore` command in SunOS release 4.x. You will be able to restore all of your old backups made with the SunOS release 4.x `dump` command, with one exception: you cannot restore multi volume backups from diskette. If you have backup scripts that invoke `restore`, change them to invoke `ufsrestore` instead.

The `dd` Command

In the SunOS release 4.x version of the `dd` command, the size suffix `w` (words) denotes a size unit of 4 bytes. In the SunOS release 5.5 version, `w` denotes a unit of 2 bytes. In addition, the SunOS release 5.5 version now supports the `unblock` and `block` conversion options.

`tar` and `cpio` Commands

Because they use a nonbinary format, the `tar` and `cpio` commands are the only utilities to successfully interchange data between SVR4 implementations. Other backup utilities, such as `ufsdump` and `dd`, are unique to the vendor, and are not guaranteed to work successfully from one SVR4 implementation to another.

The `tar` command is unchanged in this release; it accepts the same options and command syntax as the SunOS release 4.x command. However, since the device naming scheme has changed in SunOS release 5.5, the *tarfile* (or *device*)

argument is affected. When using the *f* function modifier, specify the device argument as */dev/rmt/unit*, where *unit* is a tape drive number and density. Table 9-16 shows the tape drive density characters in tape device names.

Table 9-16 Tape Drive Density Characters in Tape Device Names

Density	Description
Null	Default “preferred” (highest) density
l	Low
m	Medium
h	High
c	Compressed
u	Ultra

The `tar` command no longer uses `/dev/rmt8` as its default output device. When the `f` modifier is not used, and the `TAPE` environment variable is not set, the `tar` command uses the defaults set in the `/etc/default/tar` file.

The SunOS release 5.5 `cpio` command supports the SunOS release 4.x options and command syntax. `cpio` has been expanded to include many new options as shown in Table 9-17.

Table 9-17 `cpio` Options Added Since SunOS Release 4.x

Option	Command Available With Option ¹	Description
-A	<code>cpio -o</code>	Appends files to an archive
-k	<code>cpio -i</code>	Attempts to skip corrupt file headers and I/O errors encountered. This option lets you copy files from a medium that is corrupted or out of sequence.
-L	<code>cpio -o</code> or <code>cpio -p</code>	Follows symbolic links
-V	<code>cpio -i</code> , <code>cpio -o</code> , or <code>cpio -p</code>	Special verbose. Prints a dot for each file read or written. This option assures you that <code>cpio</code> is working, without printing all file names.

Table 9-17 `cpio` Options Added Since SunOS Release 4.x (Continued)

Option	Command Available With Option ¹	Description
<code>-C bufsize</code>	<code>cpio -i</code> or <code>cpio -o</code>	Blocks I/O <i>bufsize</i> bytes to the record, where <i>bufsize</i> is a positive integer. When neither <code>-C</code> nor <code>-B</code> is specified, the default buffer size is 512 bytes.
<code>-E filename</code>	<code>cpio -i</code>	Specifies and inputs file containing a list of file names to be extracted from the archive
<code>-H header</code>	<code>cpio -i</code> or <code>cpio -o</code>	Reads or writes header information in <i>header</i> format. <i>header</i> can be one of: <i>bar</i> (read only), <i>crc</i> , <i>CRC</i> , <i>odc</i> , <i>tar</i> , <i>TAR</i> , <i>ustar</i> , or <i>USTAR</i> .
<code>-I filename</code>	<code>cpio -i</code>	Reads <i>filename</i> as an input archive
<code>-M message</code>	<code>cpio -i -I filename</code> or <code>cpio -o -O filename</code>	Define a message to use when switching media
<code>-O filename</code>	<code>cpio -o</code>	Directs the output to <i>filename</i>
<code>-R userid</code>	<code>cpio -i</code> or <code>cpio -p</code>	Reassigns ownership and group information for each file to <i>userid</i>

1. `cpio` requires one of three mutually exclusive options to specify the action to take: `-i` (copy in), `-o` (copy out), or `-p` (pass).

Setting Up a Solaris 2.5 Server to Support SunOS Release 4.x Diskless/Dataless Clients

10 

This chapter outlines how to set up a Solaris 2.5 system as a server for SunOS release 4.x diskless/dataless clients by using the `discover4x`, `install4x`, and `convert4x` programs.

Make sure you have read Chapter 3, “Installation and Configuration Changes,” if you are setting up a Solaris 2.5 server for SunOS release 4.x clients on a Solaris 2.5 network.

Adding SunOS Release 4.x Support to a Solaris 2.4 Server

This section explains how to prepare a Solaris 2.4 server so it can serve SunOS release 4.x diskless and dataless clients.

Note – Ensure that all system data has been restored before you use the commands in this procedure. The `/export` file system is particularly important because it contains client information. See Chapter 3, “Installation and Configuration Changes.”

Some sites will need to continue using SunOS release 4.x clients after the server has been upgraded to Solaris 2.5 software. For instance, Sun 3 systems cannot run Solaris 2.2 or later and must continue to use SunOS release 4.x.

When a SunOS release 4.x `/export` partition is set up on a server running Solaris 2.5 software, it is referred to as *multiple OS operation*. Multiple OS operation enables the server to continue serving SunOS release 4.x clients while it runs the Solaris 2.5 operating environment.

The multiple OS operation package is called `SUNWhinst` and includes three programs, which you will need to run to set up a SunOS release 4.x `/export` directory on a Solaris 2.4 server. The three programs are:

- `discover4x` – This program analyzes the support that remains for SunOS release 4.x clients after the server has migrated to the Solaris 2.5 operating environment. The program looks at the SunOS release 4.x client support and creates the databases that are required for installation of SunOS release 4.x diskless/dataless clients on the Solaris 2.5 server. If client support for a given architecture is missing, `discover4x` attempts to notify users that they will have to re-install this support using `install4x`. If there are SunOS release 4.x clients with the same architecture as the server that migrated to the Solaris 2.5 operating environment, you must re-install that architecture using the `install4x` command.
- `install4x` – This program is used to install the components of a SunOS release 4.x system required to support diskless/dataless clients that existed before the migration to the Solaris 2.5 operating environment.
- `convert4x` – This program updates the Solaris 2.5 server with information about all the existing SunOS release 4.x clients. This command is used after issuing the `discover4x` and `install4x` commands. The updated information enables the existing SunOS release 4.x clients to work with the Solaris 2.5 server.

Before beginning any of these installation procedures, ensure that the `SUNWhinst` package is properly loaded. Use the `pkginfo(1)` command to generate a list of installed packages and then check the list to ensure that all necessary packages were installed, including the `SUNWhinst` package.

For details on adding and removing packages, see *System Administration Guide, Volume I*.

Running `discover4x`

`discover4x` analyzes the support that remains for SunOS release 4.x clients after the server has migrated to the Solaris 2.5 operating environment.

As superuser (root), type the following.

```
# discover4x
```

The `discover4x` program runs from 1 - 60 seconds, depending on the amount of software examined.

`discover4x` may report messages such as the following.

```
Setting up proto root for sun4c arch

Updating server databases to include sun4c sunos 4.1.2 support

Support for sun4c clients must be added using install4x, if \
sun4c clients are served by this machine.
```

If your site has completed a custom Solaris 2.5 installation that changed the location of the `/export` directory, `discover4x` examines that directory if you invoke it with the directory name as a single argument. For instance, if the `/export` software is stored in `/clients` directory, use the following command.

```
# discover4x /clients
```

Setting Up the CD-ROM Drive for install4x

Run the `install4x` program on a server with the Solaris 2.5 operating environment using one of the three procedures listed in the following section.

- “Using a Local CD-ROM Drive” if the system has a local CD-ROM drive
- “Using a Remote CD-ROM Drive (Solaris 2.4)” if the system will use a remote CD-ROM drive on a system running the Solaris 2.5 operating environment
- “Using a Remote CD-ROM Drive (SunOS release 4.x)” if the system will use a remote CD-ROM drive on a system running SunOS release 4.x

Insert the SunOS release 4.x CD into the CD-ROM drive before you proceed.

Using a Local CD-ROM Drive

If you are running `install4x` on a system with a local CD-ROM drive, after you install the CD into the drive, Volume Management automatically mounts the CD directory on `/cdrom/volume1/s0`.

Using a Remote CD-ROM Drive (Solaris 2.4)

If `install4x` is to use a CD-ROM drive on a remote system running the Solaris 2.5 operating environment, after you install the CD into the drive, Volume Management automatically mounts the CD directory on `/cdrom/volume1/s0`. Then execute the following command.

```
# share -F nfs -o ro /cdrom/volume1/s0
```

If you are not sharing other NFS systems at boot time, you need to invoke the `mountd(1M)` and `nfsd(1M)` daemons.

Type the following commands on the local system.

```
# mkdir /cdrom
# mount -F nfs -o ro cd-host:/cdrom/volume1/s0 /cdrom
```

Using a Remote CD-ROM Drive (SunOS Release 4.x)

If `install4x` is to use a CD-ROM drive on a remote system that is running SunOS release 4.x, type the following as superuser on the remote system.

```
# mkdir /cdrom
# mount -t hfs -r /dev/sr0 /cdrom
```

Once you have typed the previous commands, edit the `/etc/exports` and insert the following line.

```
/cdrom -ro
```


Then type the following command on the remote system.

```
# exportfs /cdrom
```

Type the following commands on the local system.

```
# mkdir /cdrom
# mount -F nfs -o ro cd-host:/cdrom /cdrom
```

Running install4x

After you use one of the previous procedures, the CD is mounted on /cdrom. Now invoke `install4x` by typing the following.

```
# /usr/sbin/install4x -m /cdrom/volume1/s0 -e /export
```

If the `-m` option is not specified, the following prompt is displayed.

```
Enter name of directory where the 4.1* cd is mounted [/cdrom]:
```

If the `-e` option is not specified, the following prompt is displayed.

```
Enter name of export directory [/export]:
```

As before, if your site has customized the location of the /export directory, you can direct `install4x` to load software to a different directory by specifying additional arguments as in the following.

```
# /usr/sbin/install4x -m /cdrom -e /clients
```

Choosing Software to Load

install4x displays the Install Main Menu shown here.

```
*** 4.1* Install Main Menu ***

Choose an Architecture (then select modules to load):

                Modules
                Loaded  Selected
[a] sun4.sun4c.sunos.4.1.2      8      0
[b] sun4.sun4.sunos.4.1.2      8      0
[c] sun4.sun4m.sunos.4.1.2     7      0

or begin the loading process for all selected modules:

[L] Load selected modules

or abort without loading any modules:

[Q] Quit without loading

Type any bracketed letter to select that function.

Type ? for help.
```

```
+-----+
| Disk Usage: |
|      0K Selected |
| 53634K Free |
+-----+
```

The Install Main Menu screen presents several options. The first set (labeled here as a, b, and c) is used to specify the architecture for which software is to be loaded. Other options enable the user to direct software loading to begin (L), quit the program (Q), or ask for help (?).

After you choose each appropriate architecture, the program displays the Module Selection.

```

Select sun4.sun4c.sunos.4.1.2 modules:
+[a] R proto root.....240K | [o] User_Diag.....6352K
+[b] R usr.....26240K | [p] Manual.....7456K
+[c] R Kvm.....4832K | +[q] D TLI.....48K
+[d] R Install.....936K | [r] D RFS.....912K
[e] D Networking.....1040K | [s] D Debugging.....2928K
[f] D System_V.....4008K | [t] SunView_Programmers.....1840K
[g] D Sys.....5288K | [u] Shlib_Custom.....1376K
[h] C SunView_Users.....2664K | [v] Graphics.....1784K
[i] SunView_Demo.....512K | +[w] uucp.....608K
+[j] Text.....712K | +[x] Games.....3136K
[k] Demo.....4264K | [y] Versatec.....5960K
[l] C OpenWindows_Users.....25936K | [z] Security.....312K
[m] C OpenWindows_Demo.....4288K | [A] OpenWindows_Programmers.10200K
[n] C OpenWindows_Fonts.....7840K |

Module + = already loaded R = Required C= Common
Legend: ** = selected for loading D = Desirable Others are optional

Select [a-A] or a Quick-Pick Option:
[1] All Required Modules [4] All Optional Modules | Disk Usage: |
[2] All Desirable Modules [5] All Modules | 0K Selected |
[3] All Common Modules | 53634K Free |
or [D] (done) to return to the main screen +-----+

```

Packages already loaded are shown on the Module Selection screen with a + before the selection letter (that is, in the previous screen the packages associated with letters a, b, c, d, j, q, w, and x are already loaded). Note that loading packages for one architecture may cause those packages to show as being loaded for other architectures since many packages are shared.

Select modules to load by typing the associated character that is shown in brackets. Pressing the key associated with a module toggles the selection status (that is, will select or deselect the module, depending on its previous status). Modules selected to be loaded have ** displayed before the selection character. You can reload modules already present by answering Y or y when asked to confirm the apparent redundancy.

SunSoft has determined which software must be loaded for a release to operate normally (shown with R to the right of the selection letter), which software is commonly loaded (shown as C), and which software should be loaded (shown as D).

Additionally, the Module Selection screen readily enables you to pick groups of modules to be loaded. When you enter a 1, it marks all required modules for loading. When you enter a 2, it marks all recommended modules. When you enter a 3, it marks all commonly loaded modules. When you enter a 4, it marks all optional modules. When you enter a 5, it marks all modules shown on the Module Selection screen.

Return to the Install Main Menu by typing D.

```
*** 4.1* Install Main Menu ***

Choose an Architecture (then select modules to load):

                Modules
                Loaded  Selected
[a] sun4.sun4c.sunos.4.1.2      8      0
[b] sun4.sun4.sunos.4.1.2      8      0
[c] sun4.sun4m.sunos.4.1.2     7      0

or begin the loading process for all selected modules:

[L] Load selected modules

or abort without loading any modules:

[Q] Quit without loading

Type any bracketed letter to select that function.

Type ? for help.
```

```
+-----+
| Disk Usage:      |
|           0K Selected |
|       53634K Free  |
+-----+
```

By typing L on the Install Main Menu, you can load all selected modules. Output similar to the following is displayed.

```
Installing module 'proto root' [size: 248K]
      in directory /export/exec/proto.root.sunos.4.1.2 ...

Updating server databases ...

Press any key to continue:
```

Running `convert4x`

`convert4x` updates the Solaris 2.4 server with information about all SunOS release 4.x clients. The following files and directories are updated when you run `convert4x`:

- `/tftpboot` - Directory containing network bootable images
- `/etc/dfs/dfstab` - File containing file systems exported via NFS
- `/etc/inet.conf` - File containing list of servers that `inetd(1M)` invokes when it receives an Internet request
- `/etc/bootparams` - File containing per-client boot specifications
- `/etc/hosts` - File containing IP-to-host name mapping

The following steps detail how to run `convert4x`:

1. **Make certain that the Ethernet addresses are entered in the `/etc/ethers` file for the clients you are converting. This is necessary because `convert4x` invokes the `rpc.rarpd(1m)` daemon.**
2. **Run `convert4x` by typing the following.**

```
# /usr/sbin/convert4x
```

Optionally, you can specify a single fully qualified path to the location to an alternate client hierarchy. By default, `convert4x` looks in `/export`.

As `convert4x` runs, it displays information on the screen about the actions taken by the script. It warns you if there are any discrepancies in client information. If there is insufficient information for a given client, `convert4x` reports the error and exits.

If the `convert4x` is successful for existing clients, you do not have to re-add them using Solstice Host Manager.

Setting Up and Using Printers

This chapter describes how to set up and administer printers after you install Solaris 2.5 software. This chapter also describes the changes to printer commands that have taken place between the SunOS release 4.x and the Solaris 2.5 release.

<i>Summary of Printing Differences</i>	<i>page 117</i>
<i>Print Commands and the Compatibility Package</i>	<i>page 118</i>
<i>Using Printer Commands</i>	<i>page 118</i>

Summary of Printing Differences

The SunOS release 5.5 LP print service replaces the SunOS release 4.x printing facilities, which were provided by the `lpd` daemon and `lpr`, `lpq`, `lprm`, and `lpc` commands. `Admintool` lets you set up and administer printers through a graphical user interface. You can also use a command line interface for the LP print service to administer SunOS release 5.5 printers. For detailed information about `Admintool` and the command-line interface to the LP service, see *System Administration Guide, Volume I*.

The services provided by the `/etc/printcap` file in SunOS release 4.x are handled in the Solaris 2.5 operating environment by the `terminfo` database and by the files in the `/etc/lp` directory.

Print Commands and the Compatibility Package

You can still use many SunOS release 4.x print commands if the system is running the SunOS/BSD Source Compatibility Package. Compatibility mode uses SunOS release 4.x command names as an interface to underlying Solaris 2.5 LP print services and does not actually run them the way a SunOS release 4.x system would. When a user types SunOS release 4.x commands to set up printing or to print files from a Solaris 2.5 system, the commands create message files that are handled by the SunOS release 5.5 LP print service scheduler.

Solaris 2.5 printing provides additional capabilities not available in SunOS release 4.x systems. These capabilities enable you to control forms, print wheels, and interface programs, and to set up network print services.

Using Printer Commands

As discussed in a previous section, you can continue to use SunOS release 4.x print commands if you have the SunOS/BSD Source Compatibility Package. Table 11-1 shows the basic user print command equivalents.

Table 11-1 User Print Command Equivalents

SunOS Release 4.x	SunOS Release 5.5	Function
<code>lpr filename</code>	<code>lp filename</code>	Print a file to the default printer
<code>lpr -Pprinter filename</code>	<code>lp -d printer file</code>	Print a file to a specific printer
<code>lpq</code>	<code>lpstat -o printer</code>	Look at a list of the files waiting to print on the default printer
<code>check /etc/printcap</code>	<code>lpstat -d</code>	Determine which is the default printer
<code>check /etc/printcap</code>	<code>lpstat -a</code>	Determine which printers are available
<code>lprm jobnumber</code>	<code>cancel jobid</code>	Cancel a print job on the default printer

Using SunOS 5.5 Printer Administration Commands

This section describes differences between printer setup and administration on SunOS release 4.x and Solaris 2.5 systems. All the underlying system services described are available only in the Solaris 2.5 operating environment. The SunOS release 4.x counterparts are not available even in compatibility mode.

You must use the System V printer administration commands, `lpadmin(1M)` and `lpssystem(1M)` instead. Use the `terminfo` database and the configuration files in the `/etc/lp` directory instead. See *System Administration Guide, Volume II* for details.

Table 11-2 shows the command equivalents for setting up printing.

Table 11-2 Printer Administration, Setup, and File Equivalents

SunOS Release 4.x	SunOS Release 5.5	Function
<code>lpc</code>	<code>lpadmin</code>	Control line printer functions
<code>/etc/printcap</code>	<code>terminfo</code> database and <code>/etc/lp/printers/ printername/*</code>	File that defines printer functions
<code>/var/spool</code>	<code>/var/spool/lp</code>	Directory where printing system stores spool and lock files
Not available	<code>lpmove</code>	Move print queues between printers
<code>lpc down</code>	<code>reject</code>	Stop queueing to a printer

Printing `troff`

In SunOS release 4.x, the following command sent your file to the default printer.

```
% troff filename
```

In the Solaris 2.5 operating environment, you must specify that you want the file printed by piping (|) the output to the `lp` command. Table 11-3 shows the SunOS release 5.5 `troff` commands.

Table 11-3 SunOS Release 5.5 `troff` commands

SunOS 5.2 Command	Function
<code>troff file /usr/lib/lp/postscript/dpost lp</code>	Sends to default printer that supports <code>troff</code> jobs
<code>troff file /usr/lib/lp/postscript/dpost lp -dprinter</code>	Sends to a particular printer
<code>troff file lp -Ttroff</code>	Sends to any printer that supports <code>troff</code> jobs

Managing Terminals and Modems

12 

This chapter describes serial port management (which enables terminal and modem connections) by using Admintool or the Service Access Facility (SAF).

This chapter contains the following sections:

<i>Terminal and Modem Management</i>	<i>page 121</i>
<i>Service Access Facility (SAF)</i>	<i>page 122</i>
<i>SAF Functions and Related Programs</i>	<i>page 124</i>

System Administration Guide, Volume II describes the details of Solaris 2.5 setup and installation procedures for serial devices.

Terminal and Modem Management

Admintool is a tool that readily enables you to set up and modify serial port software for terminals and modems. It provides templates for common terminal and modem configurations and multiple port setup, modification, or deletion.

Admintool features:

- Templates for common terminal and modem configurations
- Multiple port setup, modification, or deletion
- Quick visual status of each port

This tool provides the capabilities of the Service Access Facility's `pmadm` command.

Service Access Facility (SAF)

Using SAF, you can manage access to all services in a similar way, whether they are on the network or attached only to local systems. SAF uses Service Access Control (SAC) commands to set up and manage services. It provides uniform access to system services, such as:

- Adding, removing, and modifying terminal line settings
- Adding, enabling, disabling, or removing a port monitor
- Printing information from administrative database files
- Using and administering port monitors
- Adding, enabling, disabling, and removing `listen(1M)` port monitors

In previous versions of SunOS operating systems, the method for controlling devices depended both on the device providing the access, and on the location of that device. Managing user access involved editing many device files.

SAF helps isolate the system administrator from these device dependencies, and provides a common interface for managing a range of services, including the ability to:

- Log in (either locally or remotely)
- Access printers across the network
- Access files across the network

SAF's common interface consists primarily of two commands: `sacadm` and `pmadm`. The `sacadm` command controls daemons called *port monitors*. The `pmadm` command controls the services associated with the port monitors.

What's a Port Monitor?

A port monitor is a program that continuously monitors for requests to log in or requests to access printers or files.

Once a port monitor detects a request, it sets whatever parameters are required to establish communication between the operating system and the device requesting service. Then the port monitor transfers control to other processes (for example, the `login` program) that provide the services needed.

There are two types of port monitors included in the Solaris 2.5 operating environment: `ttymon` and `listen`. The `listen` port monitor controls access to network services and handles remote print and file system requests. The `ttymon` port monitor provides access to the login services needed by modems and alphanumeric terminals.

`sacadm` *Command*

The `sacadm` command lets you add and remove port monitors. You can also use the `sacadm` command to list the status of a port monitor, and to administer configuration scripts that customize port monitors.

`pmadm` *Command*

Using the `pmadm` command, you can add or remove a service, and enable or disable a service. You can, for example, disable all remote logins with one `pmadm` command. You can also install or replace per-service configuration scripts, or display information about a service.

SAF Functions and Related Programs

Using only the `sacadm` and `pmadm` commands, a system administrator has complete control over access to resources. However, these two commands are only the interface to the SAF suite of programs and processes that make the integrated management environment possible. Table 12-1 shows the functions and associated programs.

Table 12-1 SAF Functions and Associated Programs

Function	Program	Description
Overall Administration	<code>sacadm</code>	Command for adding and removing port monitors
Port Monitor Service Administrator	<code>pmadm</code>	Command for controlling port monitors' services
Service Access Control	<code>sac</code>	The SAF master program
Port Monitors	<code>ttymon</code> <code>listen</code>	Monitors serial port login requests Monitors requests for network services
Services	<code>logins</code> ; remote procedures	Services to which the SAF provides access

The service access control, `sac`, is the most important program in the SAF suite. It is launched by the `init` program when a machine is first started. In turn, `sac` starts all the port monitors listed in its administrative file.

The other processes that make up the SAF include the `ttymon` and `listen` port monitors, as well as the programs, such as `login`, that provide the various services mediated by the SAF.

For more information on the SAF in general, or on the different ways to use the `sacadm` and `pmadm` commands, see *System Administration Guide, Volume II*.

This chapter outlines changes to the network facilities, TCP/IP and UUCP.

It contains the following sections:

<i>Changes to TCP/IP</i>	<i>page 125</i>
<i>Changes to NFS</i>	<i>page 126</i>
<i>PPP</i>	<i>page 126</i>
<i>UUCP</i>	<i>page 126</i>

Changes to TCP/IP

The user interface to TCP/IP is virtually the same, but the administration of NIS+ maps is handled through the Administration Tool, which is different from the process in SunOS release 4.x and traditional AT&T SVR4.

The NIS+ maps administered by the Administration Tool include :

- Hosts
- Services
- RPC
- Ethers

When you are ready to use SunOS release 5.5 TCP/IP facilities, see *TCP/IP and Data Communications Administration Guide* for complete information.

Changes to NFS

The Solaris 2.5 operating environment simplifies resource sharing with a new set of commands and files to administer NFS resources. This set of commands is called distributed file system (DFS) administration; it replaces the NFS commands that SunOS release 4.x uses.

PPP

PPP for Solaris 2.5 systems is an asynchronous implementation of the standard data link-level, point-to-point protocol (PPP) included in the internet protocol suite. PPP enables a network administrator to create a communications link using modems and telephone lines. See *TCP/IP and Data Communications Administration Guide* for detailed information.

UUCP

The Solaris 2.5 UNIX-to-UNIX Copy (UUCP) is similar to the HoneyDanBer UUCP available with SunOS release 4.x systems. It uses the same set of configuration files, scripts, and commands, so you should be able to restore most changes you made in SunOS release 4.x files and scripts to run with this release. However, the spool directory is organized differently in Solaris 2.5 due to *job grades*, a mechanism to help sort and prioritize the work load.

Table 13-1 describes the new files and commands offered with Solaris 2.5 UUCP that were not part of the SunOS release 4.x implementation. Table 13-2 on page 129 describes the log files added to Solaris 2.5 UUCP.

Table 13-1 New SunOS Release 5.5 UUCP Files and Commands

Command or File	Description
D. data files P. data files	These data files are created when a UUCP command line specifies copying the source file to a spool directory. All data files have this format: <i>systemxxxxyyy</i> . <i>system</i> is the first five characters in the name of the remote system. <i>xxxx</i> is a four-digit job sequence number assigned by UUCP. <i>yyy</i> is a subsequence number used to distinguish between several D. files created for a work (C.) file.
<code>/etc/uucp/Grades</code>	Maps text grade names to system names.
<code>/etc/uucp/Limits</code>	Specifies the number of concurrent UUCP sessions that can occur. Replaces <code>Maxuuscheds</code> and <code>Maxuuxqts</code> files in previous versions.
<code>/etc/uucp/Config</code>	Contains information to override UUCP parameters that can be tuned. Currently, the only parameter of this type is <code>Protocol</code> , so system administrators normally will not have to modify this file.
<code>uuglist</code>	Prints the list of service grades available on the system to use with the <code>-g</code> option of <code>uucp(1C)</code> and <code>uux(1C)</code> .

Solaris 2.5 UUCP includes a few additional features that can affect system administration.

- Checkpoint-restart facilities
- Job grades that control UUCP transmission
- Two new configuration files to limit the number of concurrent UUCP sessions that the system can run, and to override UUCP parameters that can be tuned

The following sections describe the system administration differences made by each of these additions.

Checkpoint Restart

When communication link failures interrupt UUCP transmissions between SunOS release 4.x systems, the transmission starts again from the beginning of the file as soon as communication resumes. Communication between two systems running Solaris 2.5 UUCP resumes where it was interrupted, instead of restarting at the beginning. This makes better throughput possible, especially on erratic or noisy transmission lines.

The systems use two new files to store sent and received data and to compare the sizes of the files to determine where to restart transmission. The systems use `.R` files to store received data and `.D` files to store transmitted data. These files replace the `TM.` files of previous UUCP versions. If only one system is running SunOS release 5.5 UUCP, no comparison can take place and transmissions restarts from the beginning.

User Job Grades

Job grading lets administrators divide jobs into work loads that compete against others of similar size, type, priority, or all three. You can sort work loads using any one, or a combination of these factors. You can also set access permissions allowing users and groups to obtain each grade of UUCP service.

In SunOS release 4.x, the user has to choose the grade when the job is submitted. Grades are a single letter, not a name, as they are in the Solaris 2.5 operating environment. Solaris 2.5 systems enable administrators to define job grades for an entire site.

Limits File

The `/etc/uucp/Limits` file specifies the maximum number of concurrent `uucico`, `uuxqt`, and `uusched` processes permitted on a system. This single file replaces the `Maxuusched` and `Maxuuxqt` parameters on previous releases.

Config *File*

The `/etc/uucp/Config` file contains information to override UUCP parameters that can be tuned. Currently the only parameter available is `Protocol` and should normally not be altered by system administrators.

Log *Files*

Solaris 2.5 UUCP provides four log files in addition to the four supplied in previous versions. These files record accounting, command, performance, and security information. The command and security log files are created if they do not exist. The accounting and performance log files are written only if they already exist.

Table 13-2 New SunOS Release 5.5 UUCP Log Files

File Name	Function
<code>/var/uucp/.Admin/account</code>	Records account information for billing
<code>/var/uucp/.Admin/perflog</code>	Records statistics on <code>uucico</code> operations
<code>/var/uucp/.Admin/security</code>	Records attempted security violations
<code>/var/uucp/.Admin/command</code>	Records information on commands issued by users or administrators

When you are ready to set up and use SunOS release 5.5 UUCP, see *TCP/IP and Data Communications Administration Guide* for complete information.

Using Name Services

The network information service (NIS), which is part of the SunOS release 4.x environment, is widely being replaced with the *network information service plus* (NIS+). NIS+, introduced with the SunOS 5.0 system, is a completely redesigned name service that takes into account changes in customer client/server environments. DNS (domain name system) is an existing, complimentary name service used for inter-company communication. This chapter discusses NIS+ and compares it to NIS and DNS.

This chapter contains the following sections:

<i>Name Service Switch</i>	<i>page 132</i>
<i>NIS+</i>	<i>page 132</i>
<i>DNS</i>	<i>page 132</i>
<i>DNS and NIS+ Comparison</i>	<i>page 133</i>
<i>NIS and NIS+ Comparison</i>	<i>page 134</i>
<i>Planning an NIS+ Upgrade</i>	<i>page 135</i>
<i>Installing NIS+ on a SunOS Release 4.x System</i>	<i>page 135</i>

For more information about planning an NIS+ upgrade and installing NIS+, see *NIS+ Transition Guide*, *NIS+ and DNS Setup and Configuration Guide*, and *NIS+ and FNS Administration Guide*.

Note – The system administration documentation set for the Solaris 2.5 operating environment emphasizes a system that is using NIS+.

Name Service Switch

The Solaris 2.5 operating environment uses standard naming interfaces (for example, `gethostbyname`) to support multiple naming services (such as NIS, NIS+, and DNS, among others), thereby allowing applications to access data transparently from different services. One instance of this is the *Name Service Switch* capability in the Solaris 2.5 operating environment, which allows applications to use a UNIX standard naming interface (for example, `getxxbyyy` interfaces) in this manner. See the `nsswitch.conf(4)` man page for more information.

NIS+

NIS+ is a name service built on top of the ONC transport-independent remote procedure call (TI-RPC) interface. NIS+ has significant advantages over NIS in the areas of security, performance, scalability, and administration.

DNS

DNS supports the model of a hierarchical name space with autonomously administered name servers. Although NIS+ uses a similar hierarchical naming model, it focuses on supporting changing system administration data and other requirements of enterprise networks.

DNS and NIS+, therefore, are complementary name services:

- DNS is used for intercompany communication
- NIS+ supports administration of enterprise networks

DNS and NIS+ Comparison

Table 14-1 shows the features and benefits of DNS compared to NIS+.

Table 14-1 DNS and NIS+ Features and Benefits Compared

Feature	DNS	NIS+
Security	Unrestricted access to data	All operations can be optionally authenticated
API and human interface	Allows read-only access to name service	UNIX-style access rights for objects and entries Allows read-write access to name service. Provides: <ul style="list-style-type: none"> - Efficient support of changing network environment - API support of administrative operations - Support of administrative and other distributed applications
Updating	By transfer of zone master files - Weaker consistency	By incremental data transfer - Fast support of changing network environments
Compatibility with NIS	Not applicable	Existing NIS applications can migrate smoothly
Data support	ASCII data only with packet size restriction	Binary and ASCII data. Provides: <ul style="list-style-type: none"> - Support of variable information - Support of larger objects

The main strength of DNS is supporting hierarchical database partitions and replicas containing entries of relatively static information (such as host name and IP address). DNS ensures compatibility and smooth operation in a wide-area interorganization environment.

NIS+, in contrast, is a secure repository of changing administrative information (such as email aliases, Ethernet addresses, RPC program numbers) for enterprise networks.

NIS and NIS+ Comparison

Table 14-2 summarizes several major enhancements in NIS+ compared to NIS.

Table 14-2 NIS and NIS+ Features Compared

Feature	NIS	NIS+
Name space	Has a flat organization; is not hierarchical	Has a hierarchical organization
Database	Centralized flat file database for each independent network domain	Partitioned into directories to support each network subset or autonomous domain
Data Storage Scheme	Multiple bicolumn “maps” (files) having key-value pairs	Multicolumn database with multiple, searchable columns
Resource Access Across Domains	Not supported	Permitted for authorized users
Replication	One replica server required per IP subnetwork	Each replica server can serve clients on multiple IP subnets
Privileges for Updating	Updates require superuser privileges on master server	Updates can be performed remotely by authorized users
Update Process	Updates require using “make” files on master servers	Updates are performed easily through command-line interface
Update Propagation	Is administrator initiated and requires transfer of whole maps	Automatic and high-performance updating via incremental transfer
Authorization	Anyone can read all information stored in an NIS database	Fine-grained access control to NIS+ directories, table column, and entries

NIS+ includes features that enable NIS sites to migrate to the new name service in a smooth, phased manner. NIS sites that migrate to NIS+ will gain the following benefits:

- Distributed and remote administration of network domains by authorized users
- Support for hierarchical domains
- Fast and automatic propagation of updates from master to replica servers
- Fine-grained access to tables and network resources
- Easier and more consistent administrative operations
- Increased naming service reliability and availability

NIS Support in SunOS Release 5.5

The Solaris 2.5 operating environment contains support for most client-side NIS utilities and APIs. There is *no* server-side support for NIS in the Solaris 2.5 operating environment; however, NIS+ software presently includes a compatibility package (*NISBCP*) for sites that want to continue using NIS during the transition. NIS-compatibility mode makes it possible for Solaris 2.5 systems to serve both NIS+ and NIS clients.

Keep in mind, however, that NISBCP has a limited life span and will be supported only for the duration of the transition.

Planning an NIS+ Upgrade

NIS+ supports the following combinations of systems:

- SunOS release 5.5 installed on all servers and clients
- SunOS release 5.5 installed on one server, but combined with some SunOS release 4.x servers
- SunOS release 5.5 installed on some clients, running with SunOS release 4.x servers

For a network, there are three main migration paths from NIS to the NIS+ name service:

- Upgrade all servers at once to NIS+ and enable its compatibility mode (activated by the `-Y` flag of `rpc.nisd(1M)`)
- Use different domain names so NIS and NIS+ can coexist
- Have new clients use NIS+ with an NIS+ server and old clients use NIS with an NIS 4.x server, without using NIS-compatibility mode

The first step to upgrading your network is to decide which servers to upgrade to the NIS+ name service and which servers can continue to run NIS. See *NIS+ Transition Guide* for more information.

Installing NIS+ on a SunOS Release 4.x System

The Solaris 2.5 operating environment provides a way to install NIS+ on systems running earlier versions of the operating system. If you have not converted servers to Solaris 2.5 software, you can still use NIS+ to manage a heterogeneous network of SunOS release 4.x and Solaris 2.5 systems.

A `tar` archive file, `NISPLUS.TAR`, is included in the root (`/`) directory of the Solaris 2.5 distribution CD. The `NISPLUS.4.1 README` file in the root directory describes how to set up NIS+ on a SunOS release 4.x system.

Note – Sites need to upgrade to SunOS release 5.5 and then transition to NIS+.

Even though NIS+ is available for SunOS release 4.x systems, you should run it on a Solaris 2.x system. NIS+ takes advantage of many improvements that were made to non-NIS+ components in Solaris 2.x such as RPC, security, and other networking libraries.

Part 2— Transition Information for Developers

Changes in the C language and its related tools are among the most obvious differences between SunOS release 4.x and the Solaris 2.5 operating environment. These changes affect all developers to varying degrees. The operating system kernel and its interfaces have also changed significantly since SunOS release 4.x. This part of the guide describes these differences, points out similarities between releases, provides information you need to port existing software, or to write new software for the Solaris 2.5 operating environment, and explains the implications for your programming environment.

Compilers, Linkers, and Debuggers

15 

This chapter discusses the changes to compilers, linkers, and debuggers. It contains the following sections:

<i>Compilers</i>	<i>page 139</i>
<i>Linkers</i>	<i>page 146</i>
<i>Debuggers</i>	<i>page 154</i>

Compilers

The single most significant change for developers migrating from SunOS release 4.x to the Solaris 2.5 operating environment is the unbundling of the C compiler. One of the factors that allowed the compiler to be unbundled is the dynamic kernel. The compiler is no longer required to reconfigure the kernel as devices are now automatically loaded by the kernel as needed.

An ANSI C compiler is available with the unbundled SPARCworks™ product. This compiler produces executables in executable and linking format (ELF), the native object format of Solaris 2.5 executables. `lint` and the `lint` libraries are also unbundled.

Source Compatibility Guide describes the differences between the C language as implemented by the SunOS release 4.x C compiler (Sun C) and as described by the ANSI Programming Language C document (ANSI C). These differences should be addressed when porting source written for the Sun C compiler to an ANSI C conforming compiler.

Compiler Option Differences

Table 15-1 compares the SunOS release 4.x Sun C and ANSI C compilers to the Solaris 2.5 SPARCompiler C++ 3.0.1. It is provided for your information, and is not intended to describe the range of operations available in default Solaris 2.5 software.

Note – Table 15-1 presents information based on the publicly available copies of the American National Standard for Information Systems - Programming Language XX3.159-1989, approved December 14, 1989.

Please note the following:

- A “yes” in any column indicates that the option is supported by that compiler.
- If the option has changed, the new option is listed.
- A “no” in any column if the option is not supported by that driver
- A “+” in the SPARCompiler C++ 3.0.1 Solaris 2.5 column indicates an option required by SVID specifications.

Table 15-1 Comparison of C Compiler Options

Option or Flag	Sun C (SunOS release 4.x)	ANSI C (SunOS release 4.x)	SPARCompiler C++ 3.0.1 (Solaris 2.5)	Description
-A symbol	No	Yes	Yes	cpp predicate assertion
-a	Yes	Yes	-xa	Counts # basic block executions
-assertx	Yes	Yes	-z	Specifies link time assertion
-align	Yes	Yes	No	Page aligns (ld)
-BX	Yes	Yes	-d[y n]	Specifies binding type (only for libraries in SunOS release 5.5)
-bnzero	Yes	Yes	No	Generates nonzero AR
-C	Yes	Yes	+Yes	cpp comments left in
-c	Yes	Yes	+Yes	Produces .o file
-cg87	Yes	Yes	No	Sets fp option to -cg87
-cg89	Yes	Yes	No	Sets fp option to -cg89

Table 15-1 Comparison of C Compiler Options (Continued)

Option or Flag	Sun C (SunOS release 4.x)	ANSI C (SunOS release 4.x)	SPARCCompiler C++ 3.0.1 (Solaris 2.5)	Description
-dryrun	Yes	Yes	-#	Shows commands constructed by driver
-Dx	Yes	Yes	+Yes	Defines <code>cpp</code> symbol <code>x</code>
-d	Yes	Yes	No	Forces definition of common (<code>ld</code>)
-dalign	Yes	Yes	Yes	Assumes doubles are doubleword aligned
-dl	Yes	Yes	No	Generates long data segment refs
-d[y n]	-BX	-BX	Yes	Dynamics linking [yes no]
-E	Yes	Yes	+Yes	Runs source through <code>cpp</code>
-e	No	No	Yes	Entry point for <code>ld</code>
-F	-O*	-O*	+No	Optimization directives
-f	No	No	No	Floating-point support
-fast	Yes	Yes	Yes	Options for best performance
-fsingle	Yes	No	No	Floats are single precision
-fsingle2	Yes	No	No	Passes float as float not double
-fnonstd	Yes	Yes	Yes	Non-standard float option
-fstore	Yes	No	No	Forces writes on store
-G	No	No	Yes	Creates shared library, not available with the <code>-dn</code> option
-g	Yes	Yes	+Yes	Generates info for <code>dbx</code>
-go	Yes	No	No	Generates info for <code>adb</code>
-H	Yes	Yes	Yes	Prints paths of included files
-h name	No	No	Yes	Uses <code>name</code> as internal identifier; <code>soname</code> passed to linker
-help	Yes	Yes	-flags	Lists options
-Ix	Yes	Yes	+Yes	Adds <code>x</code> to include path
-J	sun3	No	No	Generates long offset for switch case

Table 15-1 Comparison of C Compiler Options (Continued)

Option or Flag	Sun C (SunOS release 4.x)	ANSI C (SunOS release 4.x)	SPARCCompiler C++ 3.0.1 (Solaris 2.5)	Description
-KPIC	-PIC	-PIC	Yes	Position independent code
-Kpic	-pic	-pic	Yes	PIC with short offsets
-Kminabi	No	No	Yes	ABI compliant code
-libmil	Yes	Yes	-xlibmil	Passes <code>libm.il</code> as part of <code>-fast</code>
-lx	Yes	Yes	Yes	Reads object library (for <code>ld</code>)
-Lx	Yes	Yes	Yes	Adds <code>x</code> to <code>ld</code> library path
-M	Yes	Yes	No	Collects dependencies
-M <i>mapfile</i>	Yes	No	Yes	Passes <i>mapfile</i> to the linker
-misalign	Yes	Yes	Yes	Handles misaligned Sun-4 data
-N	Yes	No	No	Does not make shared
-n	Yes	No	No	Makes shared
-native	Yes	Yes	No	Uses appropriate <code>-cg</code> option
-nolibmil	Yes	Yes	-xnolibmil	Doesn't pass <code>libm.il</code> with <code>-fast</code>
-noc2	Yes	Yes	No	Doesn't do peephole optimization
-o file	Yes	Yes	+Yes	Sets name of output file
-O[1,2,3,4]	Yes	Yes	-x0[1,2,3,4]	Generates optimized code
-O	Yes	Yes	+Yes	Generates optimized code
-P	Yes	Yes	+Yes	Runs source through <code>cpp</code> , output to <code>.i</code>
-PIC	Yes	No	-KPIC	Generates <code>pic</code> code with long offset
-p	Yes	Yes	+Yes	Collects data for <code>prof</code>
-pic	Yes	Yes	-Kpic	<code>pic</code> code with short offset
-pipe	Yes	No	No	Uses pipes instead of temp files
-pg	Yes	Yes	-xpg	Collects data for <code>gprof</code>
-purecross	Yes	No	No	Doesn't have slash in <code>VROOT</code>
-Qdir x	Yes	Yes	-Y*	Looks for compiler passes in <code>x</code>

Table 15-1 Comparison of C Compiler Options (Continued)

Option or Flag	Sun C (SunOS release 4.x)	ANSI C (SunOS release 4.x)	SPARCCompiler C++ 3.0.1 (Solaris 2.5)	Description
-Qpath x	Yes	Yes	-Y*	Same as -Qdir
-Qn	No	No	Yes	Doesn't add version stamp info
-Qy	No	No	Yes	Adds version stamp info
-qdir x	Yes	Yes	-Y*	Looks for compiler passes in x
-qp	-p	-p	+Yes	Collects data for prof
-ql	No	No	+Yes	Collects data for lprof
-qpath x	Yes	Yes	-Y*	Same as -Qdir
-Qoption cpp x	Yes	Yes	No	Passes option x on to program cpp
-Qoption iropt x	Yes	Yes	No	Passes option x on to program iropt
-Qoption cg x	Yes	Yes	No	Passes option x on to program cg
-Qoption inline x	Yes	Yes	No	Passes option x on to program inline
-Qoption as x	Yes	Yes	No	Passes option x on to program as
-Qoption asS x	Yes	Yes	No	Passes option x on to program asS
-Qoption ld x	Yes	Yes	No	Passes option x on to program ld
-qoption cpp x	Yes	Yes	No	Passes option x on to program cpp
-qoption ccom x	Yes	Yes	No	Passes option x on to program ccom
-qoption lint1 x	Yes	Yes	No	Passes option x on to program lint1
-qoption iropt x	Yes	Yes	No	Passes option x on to program iropt
-qoption cg x	Yes	Yes	No	Passes option x on to program cg
-qoption inline x	Yes	Yes	No	Passes option x on to program inline
-qoption cat x	Yes	Yes	No	Passes option x on to program cat
-qoption c2 x	Yes	Yes	No	Passes option x on to program c2
-qoption as x	Yes	Yes	No	Passes option x on to program as

Table 15-1 Comparison of C Compiler Options (Continued)

Option or Flag	Sun C (SunOS release 4.x)	ANSI C (SunOS release 4.x)	SPARCompiler C++ 3.0.1 (Solaris 2.5)	Description
-qoption asS x	Yes	Yes	No	Passes option x on to program asS
-qoption ld x	Yes	Yes	No	Passes option x on to program ld
-Qproduce .o	Yes	Yes	No	Produces type .o file (Object file)
-Qproduce .s	Yes	Yes	No	Produces type .s file (Assembler source)
-Qproduce .c	Yes	Yes	No	Produces type .c file (C source)
-Qproduce .i	Yes	Yes	No	Produces type .i file (C source after cpp)
-qproduce .o	Yes	Yes	No	Produces type .o file (Object file)
-qproduce .s	Yes	Yes	No	Produces type .s file (Assembler source)
-qproduce .c	Yes	Yes	No	Produces type .c file (C source)
-qproduce .i	Yes	Yes	No	Produces type .i file (C source after cpp)
-r	Yes	Yes	Yes	Makes relocatable; pass to linker
-R	Yes	Yes	No	Merges data into text segment
-R	No	No	Yes	Specifies search directories for the run-time linker
-S	Yes	Yes	+Yes	Produces .s file only
-s	Yes	Yes	Yes	Strips (4.1); pass to linker
-sb	Yes	Yes	-xsb	Collects information for code browser
-strconst	No	Yes	-xstrconst	Places string literals in read-only text segment
-sun2	Yes	No	No	Generates code for a Sun-2 system
-sun3x	Yes	No	No	Generates code for a Sun-3ax system
-sun386	Yes	No	No	Generates code for a Sun386i
-sun3	Yes	No	No	Generates code for a Sun-3 system

Table 15-1 Comparison of C Compiler Options (Continued)

Option or Flag	Sun C (SunOS release 4.x)	ANSI C (SunOS release 4.x)	SPARCCompiler C++ 3.0.1 (Solaris 2.5)	Description
-sun4c	Yes	No	No	Generates code for a Sun-4c system
-sun4	Yes	No	No	Generates code for a Sun-4 system
-target x	Yes	No	No	Sets target architecture to x
-temp=dir	Yes	Yes	No	Set directory for temps to dir
-time	Yes	Yes	No	Reports the execution times
-u	Yes	Yes	Yes	Enters symbol arg as undef (ld)
-Ux	Yes	Yes	+Yes	Undefines cpp symbol x
-v	Yes	Yes	-#	Verbose mode
-v	No	No	Yes	Strict semantic checking
-V	Yes	Yes	+Yes	Reports versions of programs
-W	No	No	+Yes	Arguments to other components
-w	Yes	Yes	Yes	Does not print warnings
-X[t,a,c,s]	No	Yes	Yes	Compatibility options
-Y	No	No	+Yes	Changes pathname to components
-yx	Yes	No	No	Traces symbol
-z	-assert	-assert	Yes	Turns on asserts in linker
-#	-v	-v	Yes	Verbose mode
EOF	No	No	Yes	File argument

The C compiler accepts the types of filename arguments shown in Table 15-2.

Table 15-2 Filename Extensions Used by the C Compiler

Suffix	File Type
.a	Object library
.il	In-line expansion file
.o	Object file
.so	Shared object

Table 15-2 Filename Extensions Used by the C Compiler (Continued)

Suffix	File Type
.s	Assembler source
.S	Assembler source for <code>cpp</code>
.c	C source
.i	C source after <code>cpp</code>

“file.X=Y” will read the file “file.X” but treat it as if it had suffix “Y”

Linkers

There are several changes to the link editor, `ld(1)`, in this release. The most important change is its ability to handle the new ELF native file format.

Note – The recommended method for building libraries and executables is through the compiler driver rather than by invoking the linker directly. The compiler automatically supplies several files needed by the linker.

Link Editor Option Differences

Some options have been renamed in the new linker, some have remained the same, and others are no longer needed. Table 15-3 compares the SunOS release 4.x `ld` command to the Solaris 2.5 `ld` command.

The sections following Table 15-3 explain how certain linking tasks are affected by the option differences.

Table 15-3 Comparison of `ld` Options

SunOS release 4.x Option	Solaris 2.5 Replacement	Notes
<code>-align datum</code>	<code>-M mapfile</code>	Uses <i>mapfiles</i> and distinct sections
<code>-assert definitions</code>	default	
<code>-assert nodefinitions</code>	<code>-znodefs</code>	Issues a fatal error instead of a warning
<code>-assert nosymbolic</code>	<code>-zdefs</code>	Issues a fatal error instead of a warning
<code>-assert pure-text</code>	<code>-ztext</code>	Issues a fatal error instead of a warning

Table 15-3 Comparison of ld Options (Continued)

SunOS release 4.x Option	Solaris 2.5 Replacement	Notes
-A <i>name</i>	No replacement	dlopen(3X) and dlclose(3X) can approximate this behavior
-Bdynamic	-Bdynamic	Applies only to the inclusion of shared libraries; use -dy (the default) to build dynamically linked executables. See "Building Executables" on page 148.
-Bnosymbolic	-zdefs	
-Bstatic	-dn & -Bstatic	Used with the -dn option to completely eliminate the dynamic linker, the -dn option must be specified; use -Bstatic in dynamic mode to include archive libraries. (Used as a toggle. See "Building Executables" on page 148.)
-Bsymbolic	-Bsymbolic	Also gets -assert nosymbolic with this option
-d -dc -dp	Default	Use -b option in SVR4 to turn off
-D <i>hex</i>	-M <i>mapfile</i>	<i>mapfiles</i> contain different mechanisms to accomplish desired effect
-e <i>entry</i>	-e <i>entry</i>	
no -e	-G	Creates a shared object
-lx[.v]	-lx	Only major number versioning of shared libraries is currently supported
-L <i>dir</i>	-L <i>dir</i>	<i>dir</i> not recorded in executable; use -R option instead.
-M	-m	
-n	Default	SVR4 executable format compresses disk image as -n
-N	No replacement	
-o <i>name</i>	-o <i>name</i>	
-p	Default	Can override with -M <i>mapfile</i>
-r	-r	

Table 15-3 Comparison of ld Options (Continued)

SunOS release 4.x Option	Solaris 2.5 Replacement	Notes
-S	No replacement	
-s	-s	
-t	no replacement	
-T <i>hex</i>	-M <i>mapfile</i>	<i>mapfiles</i> contain different mechanisms to accomplish desired effect
-Tdata <i>hex</i>	-M <i>mapfile</i>	<i>mapfiles</i> contain different mechanisms to accomplish desired effect
-u <i>name</i>	-u <i>name</i>	
-x	No replacement	
-X	No replacement	
-ysym	No replacement	
-z	default	SVR4 executable format demand pages as -z

Building Shared Libraries

The procedure for building shared libraries has changed in the Solaris 2.5 operating environment. The `-G` option is now required to build a shared library. In SunOS release 4.x, the linker would infer that a shared library was being built by the absence of the `-e` option. As shared libraries may have entry points, this option can no longer be used.

Building Executables

The `-Bdynamic` and `-Bstatic` options are still available, but their behavior is different. These options now refer to library inclusions; not to the executable binding. Executable binding is set exclusively with the new `-dy` and `-dn` options in the Solaris 2.5 software. The `-dy` option is the default. It is required to create a dynamically linked executable. The `-dn` option is required to create a statically linked executable.

The `-Bdynamic` and `-Bstatic` options apply only when using the `-dy` option. `-Bdynamic` tells the link editor to include shared libraries, while `-Bstatic` tells it to include archive libraries. These options act as a toggle governing subsequent `-l` arguments until the next `-Bdynamic` or `-Bstatic` option is encountered.

The following examples show SunOS release 4.x and Solaris 2.5 commands that can be used to create similar executables.

```
sunos4.1% ld -Bstatic test.o -lx           # uses libx.a and creates a static executable
sunos5.x% cc -dn test.o -lx              # uses libx.a and creates a static executable
sunos4.1% ld -Bdynamic test.o -lx        # uses libx.so and creates a dynamic
                                          # executable
sunos5.x% cc test.o -lx                  # uses libx.so and creates a dynamic
                                          # executable
sunos4.1% ld -Bdynamic test.o -Bstatic -lx # uses libx.a and creates a dynamic
                                          # executable
sunos5.x% cc test.o -Bstatic -lx         # uses libx.a and creates a dynamic
                                          # executable
```

Specifying Library Search Paths

In SunOS release 4.x, directories specified with the `-L` option were searched at link time and the information retained for use at execution time. This behavior is now divided between the `-L` and `-R` options. The `-L` option specifies the directories to search at link time; the new `-R` option tells the linker the search paths to be retained for use at run time. See “Search Path Rules,” in the next section for more information.

As with the `-Bdynamic` and `-Bstatic` options, the position of the `-L` option has significance; it applies only to the subsequent `-l` options.

Search Path Rules

The dynamic linker and the run time linker determine their search paths through a different algorithm from that used by the SunOS release 4.x linker.

Table 15-4 through Table 15-6 compare the search paths for the dynamic linker and the run time linker for SunOS release 4.x and the Solaris 2.5 operating environment. Notice that in the latter, the search path for the link editor and the runtime linker are affected by the LD_LIBRARY_PATH setting.

Table 15-4 SunOS Release 4.x Linker Search Paths

	Library Search Path
Link Editor	-L, LD_LIBRARY_PATH, /usr/lib, /usr/local/lib
Runtime Linker	LD_LIBRARY_PATH, -L, /usr/lib, /usr/local/lib

Table 15-5 Solaris 2.5 Linker Search Paths With LD_LIBRARY_PATH=dirlist1

	Library Search Path
	LD_LIBRARY_PATH=dirlist1
Link Editor	-L, dirlist1, /usr/ccs/lib, /usr/lib
Runtime Linker	dirlist1, -R, /usr/lib

Table 15-6 Solaris 2.5 Linker Search Paths With LD_LIBRARY_PATH=dirlist1,dirlist2

	Library Search Path
	LD_LIBRARY_PATH=dirlist1; dirlist2
Link Editor	dirlist1, -L, dirlist2, /usr/ccs/lib, /usr/lib
Runtime Linker	dirlist1, dirlist2, -R, /usr/lib

Version Numbering

SunOS release 4.x supported both major and minor version numbers on shared libraries. The Solaris 2.5 operating environment supports only the major version number. For binary compatibility support, major and minor version

numbers are recognized on SunOS release 4.x shared libraries. These libraries are required to retain the same major and minor version number they had in SunOS release 4.x.

Table 15-7 shows versions of SunOS release 4.x and Solaris 2.5 shared libraries.

Table 15-7 Example Shared Libraries

SunOS release 4.x	Solaris 2.5
libc.so.1.7	libc.so.1
libdl.so.1.0	libdl.so.1

In SunOS release 4.x system software, when the `-l` option was specified, the build environment linker searched for a library with both major and minor numbers. For example, if `-ldl` was specified, the library, `libdl.so.1.0` was linked. In Solaris 2.5, even though major numbers are still supported, the default behavior of the link editor is to ignore version numbers. Using the previous example, the build environment link editor now searches for `libdl.so` and a symbolic link points to a specific version file.

The recording of a *dependency* in a dynamic executable or shared object is, by default, the filename of the associated shared object as it is referenced by the link-editor. To provide a more consistent means of specifying dependencies, shared objects can record within themselves the filename by which they should be referenced at runtime. This is specified with the `-h` option when linking the library file.

Symbolic links have been created for most libraries in this release. Build any new shared libraries with major numbers, then create a symbolic link to the version of the library that is used most often. .

Examples

A new utility, `dump(1)` (See “Backing Up and Restoring Files” on page 101), makes it easier to debug object files or to check the static and dynamic linking. The `dump -L` option displays the information needed by the runtime linker that is contained in the executable. This information is contained in the *dynamic section* of an ELF file. The `RPATH` entry displays search paths specified by the `-R` option to `ld`.

The following example:

- Builds a shared library, `libx.so.1`, using `libx.o`
- Creates a link from `libx.so.1` to `libx.so`
- Shows `dump` output, including the `SONAME` field, which stores the information passed with the `-h` option.

```
examples% cc -G -o libx.so.1 -h libx.so.1 libx.o

examples% cp libx.so.1 /mylibs

examples% ln -s /mylibs/libx.so.1 /mylibs/libx.so

examples% dump -Lv libx.so.1

libx.so.1:

    **** DYNAMIC SECTION INFORMATION ****
.dynamic :
[INDEX] Tag      Value
[1]     INIT     0x3b8
[2]     FINI     0x3f4
[3]     SONAME   libx.so.1
[4]     HASH     0x94
[5]     STRTAB   0x33c
[6]     SYMTAB   0x14c
[7]     STRSZ    0x62
[8]     SYMENT   0x10
[9]     PLTGOT   0x10404
[10]    PLTSZ    0xc
[11]    PLTREL   0x7
[12]    JMPREL   0x3ac
[13]    RELA     0x3a0
[14]    RELASZ   0x18
[15]    RELAENT  0xc
```

If a library needs other dynamic libraries, they should be specified along with an `RPATH`, as the next example shows.

The next example compiles `prog.c`, dynamically linking `libx.so` (as built in the previous example), and specifies that the binary retain the current directory information for execution. This example shows the output of `dump` from the

compiled program, `prog.c`. Here, the information stored in the `SONAME` field of the previous example is shown as `NEEDED` by `prog`. When `prog` is run, it will use `libx.so.1`, even if `libx.so` is linked to a different version.

```
examples% cc -o prog prog.c -L/mylibs -R/mylibs -lx
example% dump -Lv prog
prog:
    **** DYNAMIC SECTION INFORMATION ****
.dynamic :
[INDEX]  Tag      Value
[1]  NEEDED  libx.so.1
[2]  NEEDED  libc.so.1
[3]  INIT     0x1b1ac
[4]  FINI     0x1b248
[5]  RPATH    /mylibs
[6]  HASH     0x100e8
[7]  STRTAB   0x17f90
[8]  SYMTAB   0x12be0
[9]  STRSZ    0x31e1
[10] SYMENT   0x10
[11] DEBUG   0x0
[12] PLTGOT  0x2b25c
[13] PLTSZ   0x30
[14] PLTREL   0x7
[15] JMPREL   0x1b180
[16] RELA    0x1b174
[17] RELASZ  0x3c
[18] RELAENT  0xc
```

Debuggers

This section describes changes to debugging tools.

`dbx` *and* `dbxtool`

These tools are no longer available with default system software. Enhanced versions of these tools are available as the SPARCworks Debugger, an unbundled product.

`adb` *and* `kadb`

These tools are available in the Solaris 2.5 operating environment. They offer the same capabilities as the tools in SunOS release 4.x. `kadb` has been enhanced to recognize multiple processors. The processor ID is displayed in the `kadb` prompt. In the following examples, it is 0.

To make kernel debugging under the Solaris 2.5 operating environment easier:

- Enable `savecore` (uncomment the `savecore` lines in the `/etc/init.d/sysetup` file)
- Boot under `kadb` (type `$c` when the system crashes)
- Use `adb` and `crash`

`kadb` *Macros*

The following `kadb` macros are particularly useful with the new multithreaded kernel:

`thread`

Displays the current thread. The current thread pointer is in SPARC global register `g7`.

```
kadb[0]: <g7$<thread
```

threadlist

Shows the stack traces of all the kernel threads in the system. This can be a *long* list.

```
kadb[0]: $<threadlist
```

mutex

Shows you the address of the owning thread, given a `mutex`. This example uses the global unsafe driver `mutex`.

```
kadb[0]: unsafe_driver$<mutex
```

moddebug

```
kadb[0]: moddebug/W 0x80000000
```

Allows you to watch module loading. See the end of `<sys/modctl.h>` for legal values for `moddebug` for debugging purposes only.

Debugging a Live Kernel

Use the following command to debug a live kernel.

```
# adb -k /dev/ksyms /dev/mem
```

`/dev/ksyms` is a pseudo device that contains the complete name list of the running kernel.

truss Command

`truss` is a new utility, provided to trace system calls performed, signals received, and machine faults incurred. `truss` offers several significant improvements over the SunOS release 4.x `trace(1)` command: including the ability to follow forked processes and the ability to deal with multithreaded processes.

The following example shows a summary of traced calls for the `date` command. With the `-c` option, `truss` does not display the trace line-by-line. Instead, it counts the system calls, signals, and faults, and displays a summary.

```
example% truss -c date
Fri Sep 18 14:31:30 PDT 1992
syscall      seconds    calls   errors
_exit        .00        1
read         .00        7
write        .00        1
open         .03       12
close        .00       12
time         .00        1
brk          .01        4
lseek        .00        1
fstat        .00        4
ioctl        .00        1
execve       .00        1
mmap         .01       17
munmap       .00        8
-----
sys totals:  .05       70      0
usr time:    .03
elapsed:    .28
```

See the `truss(1)` man page in *man Pages(1): User Commands* for complete details on all `truss` options.. In Solaris 2.5 there are a number of other new debugging tools which are based on `proc(4)` such as `pmap(1)`.

This chapter discusses the changes to tools and resources for the development environment. This chapter contains the following sections:

<i>ioctl() Requests</i>	<i>page 157</i>
<i>ptrace() Request Values</i>	<i>page 159</i>
<i>Libraries</i>	<i>page 161</i>
<i>Using make</i>	<i>page 164</i>
<i>Using SCCS</i>	<i>page 165</i>
<i>Determining Application Compatibility</i>	<i>page 165</i>
<i>Packaging Applications</i>	<i>page 166</i>
<i>Toolkits</i>	<i>page 167</i>
<i>Finding SunOS Release 4.x Tools</i>	<i>page 168</i>

ioctl() Requests

All `ioctl`s related to `dkio(7I)`, `filio`, `mtio(7I)`, `sockio(7I)`, `streamio(7I)`, `termio(7I)`, and `termios(7I)` are supported in this release.

A few incompatibilities exist between the SunOS release 4.x `termios` structure and Solaris 2.5 `termios` structure. Solaris 2.5 `termios` structure does not include a `c_line` field as it does in SunOS release 4.x. In the SunOS release 4.x,

the `c_cflag` (hardware control of the terminal) can have `CRTSCTS` (enable RTS/CTS flow control), but this value is not defined in SVR4 or the Solaris 2.5 operating environment.

Table 16-1 shows `ioctl`s defined in `<sys/ttold.h>` that are not implemented in this release.

Table 16-1 Unsupported `ioctl` Requests

TIOCMODG	TIOCMIDS	TIOCGETX
OTTYDISC	TIOCSETX	NTTYDISC
TABLDISC	NETLDISC	MOUSELDISC
KBLDISC	NTABLDISC	

Table 16-2 shows `ttycom` `ioctl` requests that are not in the Solaris 2.5 operating environment.

Table 16-2 Unsupported `ttycom` `ioctl` Requests

TIOCSCTTY	TIOCGETPGRP	TIOCSETPGRP
TIOCNOTTY	TIOCISIZE	TIOCTCNTL
TIOCISPACE	TIOCUCNTL	TIOCCONS
TIOCPKT	TIOCOUTQ	

Table 16-3 shows the `ioctl`s supported in the Solaris 2.5 operating environment.

Table 16-3 `ioctl` Support

<code>ioctl()</code>	Description
DKIOCGPART	These requests are replaced with <code>DKIOCGAPART</code> and <code>DKIOCSAPART</code> in Solaris 2.5 software.
DKIOCGCONF	This request is replaced with <code>DKIOCINFO</code> in Solaris 2.5 software, which includes the combined information of the SunOS release 4.x <code>DKIOCGCONF</code> and <code>DKIOCINFO</code> structures.
DKIOCSCMD	This request succeeds only for IPI drives. This <code>ioctl</code> fails for SCSI devices. Use the <code>USCSI</code> <code>ioctl</code> for SCSI devices.
DKIOCGLOG	<code>EINVAL</code> is returned. <code>DKIOCWCHK</code> toggles the write check on the floppy device.

Table 16-3 ioctl() Support (Continued)

ioctl()	Description
filio	The following filio ioctl requests are not supported in this release or SVR4: FIOSETOWN, FIOGETOWN, FIOCLEX, FIONCLEX. filio ioctl requests not defined in the ABI or SVID.
mtio	Not all devices support all mtio ioctl requests in Solaris 2.5. See mtio(7).
sockio	The following sockio ioctl requests are implemented in SVR4 and Solaris 2.5 software: SIOCSPGRP, SIOCGPGRP, SIOCATMARK. sockio ioctl requests are not defined in the ABI or SVID.
streamio	All SunOS release 4.x streamio ioctl requests are implemented in Solaris 2.5 software, the ABI, SVID, and SVR4. The I_FDINSERT request requires an argument that points to a strfdinsert structure. The SunOS release 4.x strfdinsert structure includes an fd (int) field, while the ABI, SVID, or SVR4 strfdinsert structure includes a fildes (int) field instead.
audioio	The SunOS release 4.x <sun/audioio.h> file has been moved to <sys/audioio.h> for Solaris 2.5 software. Additionally, in Solaris 2.5 software, there are enhancements to the interface. See audio(7), audioamd(7), or dbri(7) for more information.
termio, termios	All SunOS release 4.x termio and termios ioctl requests are implemented in Solaris 2.5 software, the ABI, SVID, and SVR4. There are a few incompatibilities between the SunOS release 4.x termios structure and Solaris 2.5 software, or the ABI, SVID, or SVR4 termios structure. The SunOS release 4.x termios structure includes a c_line field that is not supported by the other releases. The c_cflag (hardware control of the terminal) can have CRTSCTS (enable RTS/CTS flow control) under SunOS release 4.x, but this value is not defined in Solaris 2.5 software, the ABI, SVID, or SVR4. However, the functionality is supported through the termiox(7) interface.

ptrace() Request Values

The ptrace() facility is implemented on top of /proc. New applications should use proc(4) directly.

The `ptrace()` routine in Solaris 2.5 software is present solely to support applications running in BCP mode. It uses integers 1 – 9 as request values, while the SunOS release 4.x routine defines request values as symbolic constants in `<sys/ptrace.h>`. The following SunOS release 4.x request symbolic constants are compatible with Solaris 2.5 software:

`PTRACE_TRACEME`, `PTRACE_PEEKTEXT`, `PTRACE_PEEKDATA`,
`PTRACE_PEEKUSER`, `PTRACE_POKETEXT`, `PTRACE_POKEDATA`,
`PTRACE_POKEUSER`, `PTRACE_CONT`, `PTRACE_KILL`, `PTRACE_SINGLESTEP`.

The SunOS release 4.x `PTRACE_CONT` (Solaris 2.5 request 7) *addr* argument specifies where the stopped process should resume execution, unless *addr* = 1, in which case execution resumes from where the process had stopped. The Solaris 2.5 request 7 (SunOS release 4.x `PTRACE_CONT`) requires that *addr* always be equal to 1 and that execution always resumes from where the process had stopped. Also, the Solaris 2.5 request 7 cancels all pending signals, except those specified by data, before the process resumes execution. The SunOS release 4.x `PTRACE_CONT` does not cancel all pending signals.

Table 16-4 shows SunOS release 4.x valid requests that are not supported by the Solaris 2.5 `ptrace()` routine.

Table 16-4 `ptrace()` Requests not Supported by Solaris 2.5 Software

<code>PTRACE_ATTACH</code>	<code>PTRACE_GETWINDOW</code>
<code>PTRACE_DETACH</code>	<code>PTRACE_SETWINDOW</code>
<code>PTRACE_GETREGS</code>	<code>PTRACE_22</code>
<code>PTRACE_SETREGS</code>	<code>PTRACE_23</code>
<code>PTRACE_GETFPREGS</code>	<code>PTRACE_26</code>
<code>PTRACE_SETFPREGS</code>	<code>PTRACE_27</code>
<code>PTRACE_READDATA</code>	<code>PTRACE_28</code>
<code>PTRACE_WRITEDATA</code>	<code>PTRACE_SYSCALL</code>
<code>PTRACE_READTEXT</code>	<code>PTRACE_DUMPCORE</code>
<code>PTRACE_WRITETEXT</code>	<code>PTRACE_SETWRBKPT</code>
<code>PTRACE_GETFPAREGS</code>	<code>PTRACE_SETACBKPT</code>
<code>PTRACE_SETFPAREGS</code>	<code>PTRACE_CLRDR7</code>

Libraries

This release is compliant with the System V Interface Definition, Third Edition (SVID 3). Programs written with the SunOS release 4.1 System V libraries are easy to port to this release. Programs using the SunOS release 4.x BSD C library require more effort.

Reorganized Libraries

Several functions and groups of functions that were in particular libraries in SunOS release 4.x were moved into different libraries in Solaris 2.5. This can cause references to these functions to be flagged as undefined when compiling a 4.x application in Solaris 2.5.

After a compile, check the man page of any functions flagged as undefined. The synopsis list both the `-l` linker option and any include files that you need to resolve the symbol.

Shared Libraries

Shared libraries do not currently support minor version numbers.

Files for shared initialized data (`.sa`) are no longer required; no `.sa` files are provided with the Solaris 2.5 software.

Resource Limits

The Solaris 2.5 operating environment handles resource limits differently from SunOS release 4.x. In previous releases, static table allocations were used for resources such as file descriptors and active processes. These resources are now dynamically allocated. This means that they are limited by the physical memory available. Table 16-5 shows the resource limits.

Table 16-5 Resource Limits

Configuration	Limitation
RLIMIT_CORE	Maximum size of core file (in bytes) that can be created by a process
RLIMIT_CPU	Maximum amount of CPU time (in seconds) that a process can use
RLIMIT_DATA	Maximum size of a process's heap (in bytes)
RLIMIT_FSIZE	Maximum size of a file (in bytes) that can be created by a process
RLIMIT_NOFILE	One more than the maximum number of file descriptors that can be created by a process
RLIMIT_VMEM	Maximum size (in bytes) to which a process's mapped address space may grow
RLIMIT_STACK	Maximum size (in bytes) of a process's stack

Note – Any shared objects that need the networking libraries *must* be dynamically linked. The networking libraries require `libdl.so.1`. An archive library is not available.

Table 16-6 shows SunOS release 4.x and Solaris 2.5 libraries and their locations.

Table 16-6 Comparison of Library Locations

Library Name	SunOS release 4.x Directory	Solaris 2.5 Directory
<code>libbsdmalloc.a</code>	<code>/usr/lib</code>	<code>/usr/lib</code>
<code>libc.a</code>	<code>/usr/lib</code> , and <code>/usr/5lib</code>	<code>/usr/lib</code>
<code>libc.so.1.7</code>	<code>/usr/lib</code>	<code>/usr/lib</code>
<code>libc.so.2.7</code>	<code>/usr/5lib</code>	<code>/usr/lib</code>
<code>libc_p.a</code>	<code>/usr/5lib</code>	Not available
<code>libcurses.a</code>	<code>/usr/lib</code> , and <code>/usr/5lib</code>	<code>/usr/ucblib</code> , and <code>/usr/ccs/lib</code>
<code>libcurses_p.a</code>	<code>/usr/5lib</code>	Not available

Table 16-6 Comparison of Library Locations (Continued)

Library Name	SunOS release 4.x Directory	Solaris 2.5 Directory
libdbm.a	/usr/lib	/usr/ucblib
libdl.so.1.0	/usr/lib	/usr/lib
libg.a	/usr/lib	Not available
libkvm.a	/usr/lib	Not available
libkvm.so.0.3	/usr/lib	/usr/lib
libl.a	/usr/lib	/usr/ccs/lib
libln.a	/usr/lib	Not available
liblwp.a	/usr/lib	Not available
libm.a	/usr/lib	/usr/lib, and /usr/lib/libp
libmp.a	/usr/lib	/usr/lib
libnbio.a	/usr/lib	Not available
libnsl.a	/usr/lib	/usr/lib
libpixrect.a	/usr/lib	Not available
libpixrect.so.2.14	/usr/lib	Not available
libposix.a	/usr/lib	Not available
libresolv.a	/usr/lib	/usr/lib
librpcsvc.a	/usr/lib	/usr/lib
libsuntool.so.0.54	/usr/lib	Not available
libsunwindow.so.0.55	/usr/lib	Not available
libsvdm.a	/usr/5lib	Not available
libsvdm_p.a	/usr/5lib	Not available
libtermcap.a	/usr/lib, /usr/5lib	/usr/ucblib, and /usr/ccs/lib
libtermplib.a	/usr/lib, and /usr/5lib	/usr/ccs/lib

Table 16-6 Comparison of Library Locations (Continued)

Library Name	SunOS release 4.x Directory	Solaris 2.5 Directory
libxgl.so.1.1	/usr/lib	/opt/SUNWits/Graphi cs-sw/xgl/lib
libxpg.a	/usr/xpg2lib	Not available
liby.a	/usr/lib, and /usr/5lib	/usr/ccs/lib

Using make

There are two make utilities available in the Solaris 2.5 operating environment. The default version, /usr/ccs/bin/make, is identical to the SunOS release 4.x make command. The SVR4 version is available in /usr/ccs/lib/svr4-make.

Using the default version, your Makefiles will not need changes. However, some of the commands used in your Makefiles may have changed. For example, install(1), commonly used in Makefiles, could produce unexpected results because of changes to the options. (See the following code sample.)

```
In a SunOS 4.x Makefile:
install:
    install -o bin -g bin -m 444 target.c /usr/bin/target

In a SunOS 5.5 Makefile:
install:
    install -u bin -g bin -m 444 target.c /usr/bin/target
```

The version of install(1B) in /usr/ueb is compatible with the 4.x version.

Check the compatibility tables in Appendix A, “Commands Reference Table,” for information about individual interfaces.

Using SCCS

The Solaris 2.5 operating environment source code control system (SCCS) has minor differences from the SunOS release 4.x version. The same set of commands and subcommands are supported in both environments. SCCS directories and `s.files` used on SunOS release 4.x systems work equally well on Solaris 2.5 systems.

In SunOS release 4.x, the SCCS commands were located in the `/usr/sccs` directory. These commands are located with the other programming tools in `/usr/ccs/bin` in the Solaris 2.5 operating environment.

One difference between SunOS release 4.x and Solaris 2.5 utilities is the handling of unreadable `s.files`. The SunOS release 4.x commands print an error and continue when they encounter an unreadable `s.file`. The Solaris 2.5 commands silently ignore the error.

Determining Application Compatibility

Although the Binary Compatibility Package is not provided as a development environment, it requires sound programming practices that can improve binary compatibility with future releases.

Information on using the Binary Compatibility Package is available in *Binary Compatibility Guide*.

Well-Behaved Applications

The Binary Compatibility Package works with *well-behaved* user applications. Well-behaved applications meet the following requirements:

- They do not trap directly to the kernel
- They do not write directly to any system files
- They do not use `/dev/kmem`, `/dev/mem`, or `libkvm`
- They do not use unpublished SunOS interfaces
- They do not rely on customer-supplied drivers

Applications that are not well-behaved can produce unpredictable results.

Dynamically Linked and Statically Linked Executables

The Binary Compatibility Package provides compatibility for dynamically linked and statically linked applications, as well as hybrids that are partially static and partially dynamically linked.

Packaging Applications

The Solaris 2.5 operating environment is bundled in units called *packages*. These packages contain all of the files and information you need to add or remove software from your system.

A package consists of the following components:

- `pkginfo` file – This is an ASCII file that sets characteristics of the package. It consists of a list of *macro=value* pairs that describe the package and set control parameters for its installation. See the `pkginfo(4)` man page for more information.

`prototype` file – This is an ASCII file that defines the contents of the package. It contains one entry for each deliverable object (for example, files, directories, and links). It also contains installation entries for package *information* files, such as `pkginfo`, `depend`, and `copyright` files, and scripts. See the `prototype(4)` man page for more information.

`copyright` file – This is an ASCII file that provides a copyright notice for the package. Its contents (including comment lines) are displayed during package installation. See the `copyright(4)` man page for more information.

Package contents – The contents of the package.

Scripts – Scripts can be used to control installation or removal of a package, to request input from the user, or to perform an action on all objects of a particular class. Scripts must be executable by the Bourne shell.

Add-on application software should be packaged so it can be installed on a Solaris 2.5 system from diskette, tape, or CD-ROM. *Application Packaging Developer's Guide* provides guidelines for building your packages.

Packaging Utilities

Several utilities are provided to create and manipulate packages.

Table 16-7 lists commands that are useful for creating packages.

Table 16-7 Commands for Creating Packages

<code>pkgproto</code>	Generates prototype file entries for input to the <code>pkgmk</code> command
<code>pkgmk</code>	Produces an installable package
<code>pkgtrans</code>	Translates package format

Table 16-8 lists commands that are useful for adding and removing packages.

Table 16-8 Commands for Adding and Removing Packages

<code>pkgadd</code>	Add software package to the system
<code>pkgask</code>	Store answers to a request script
<code>pkgrm</code>	Remove a package from the system
<code>pkgchk</code>	Check accuracy of installation

Table 16-9 lists commands that provide information about packages.

Table 16-9 Commands for Providing Information About Packages

<code>pkginfo</code>	Display software package information about installed packages
<code>pkgparam</code>	Display package parameter values

Toolkits

This section discusses changes to OLIT and XView.

OLIT

The OPEN LOOK Intrinsic Toolkit (OLIT) is based on Xt Intrinsic. It provides a set of functions common to many widget sets to create, employ, and destroy user interface components for an X environment.

XView

The XView Window Toolkit provides an implementation of the OPEN LOOK Graphical User Interface (GUI) specification. It provides a migration path for SunView applications.

XView uses variable-length attribute-value lists based on `varargs` to specify objects to be created, such as windows, menus, and scrollbars. This eliminates most of the boilerplate software usually found in procedural interfaces, since the usual behavior is already defined.

Finding SunOS Release 4.x Tools

Most programming tools are still available in the Solaris 2.5 operating environment, and they provide the same capabilities, but many are in new locations. All bundled programming tools are now in two directories: `/usr/ccs/bin` and `/usr/ccs/lib`. Table 16-10 shows the programming tools and their locations.

Table 16-10 Bundled Programming Tools

SunOS release 4.x Command	SunOS release 4.x Location	Solaris 2.5 Location
admin	<code>/usr/sccs</code>	<code>/usr/ccs/bin</code>
ar	<code>/usr/bin</code>	<code>/usr/ccs/bin</code>
as	<code>/usr/bin</code>	<code>/usr/ccs/bin</code>
cdc	<code>/usr/sccs</code>	<code>/usr/ccs/bin</code>
comb	<code>/usr/sccs</code>	<code>/usr/ccs/bin</code>
cpp	<code>/usr/lib/cpp</code>	<code>/usr/ccs/lib</code>
delta	<code>/usr/sccs</code>	<code>/usr/ccs/bin</code>
error	<code>/usr/ucb</code>	<code>/usr/ccs/bin</code>
get	<code>/usr/sccs</code>	<code>/usr/ccs/bin</code>
help	<code>/usr/sccs</code>	<code>/usr/ccs/bin</code>
ld	<code>/usr/bin</code>	<code>/usr/ccs/bin</code>
lex	<code>/usr/bin</code>	<code>/usr/ccs/bin</code>
lorder	<code>/usr/bin</code>	<code>/usr/ccs/bin</code>
m4	<code>/usr/bin</code>	<code>/usr/ccs/bin</code>
make	<code>/usr/bin</code>	<code>/usr/ccs/bin</code>
nm	<code>/usr/bin</code>	<code>/usr/ccs/bin</code>
prof	<code>/usr/bin</code>	<code>/usr/ccs/bin</code>

Table 16-10 Bundled Programming Tools (Continued)

SunOS release 4.x Command	SunOS release 4.x Location	Solaris 2.5 Location
prs	/usr/sccs	/usr/ccs/bin
prt	/usr/sccs	/usr/ccs/bin
ranlib	/usr/bin	/usr/ccs/bin
rmdel	/usr/sccs	/usr/ccs/bin
sact	/usr/sccs	/usr/ccs/bin
sccs	/usr/ucb	/usr/ccs/bin
sccsdiff	/usr/sccs	/usr/ccs/bin
size	/usr/bin	/usr/ccs/bin
strip	/usr/bin	/usr/ccs/bin
symorder	/usr/ucb	/usr/ccs/bin
tsort	/usr/bin	/usr/ccs/bin
unget	/usr/sccs	/usr/ccs/bin
unifdef	/usr/ucb	/usr/ccs/bin
val	/usr/sccs	/usr/ccs/bin
vc	/usr/old	/usr/ccs/bin
what	/usr/sccs	/usr/ccs/bin
yacc	/usr/bin	/usr/ccs/bin
yaccpar	/usr/lib	/usr/ccs/bin

Table 16-11 lists the new Solaris 2.5 commands and their descriptions.

Table 16-11 New Programming Tools

New Command	Description
dis	Object code disassembler
dump	Dumps selected parts of an object file
exstr	Extracts strings from source files
mcs	Manipulates the comment section of an object file

Table 16-11 New Programming Tools

New Command	Description
regcmp	Regular expression compiler
truss	Traces system calls and signals
ptools	Miscellaneous <code>/proc</code> utilities

Table 16-12 lists the SunOS release 4.x commands that are now unbundled.

Table 16-12 Unbundled Programming Tools

Unbundled Command	Description
cb	A simple C program beautifier
cc	C compiler
cflow	Generates a flow graph for a C program
cscope	Interactively examines a C program
ctrace	Generates a C program execution trace
cxref	Generates a C program cross-reference
dbx	Source-level debugger
dbxtool	Window-based source-level debugger
gprof	Displays call-graph profile data
indent	Indents and formats C program source files
inline	In-line procedure call expander
lint	A C program verifier
objdump	Dumps selected parts of a COFF object file
tcov	Constructs test coverage analysis and statement-by-statement profile

Networking and Internationalization

This chapter discusses Solaris 2.5 networking features as they relate to the programming environment, and it discusses issues concerning the improved internationalization features.

<i>Networking</i>	<i>page 171</i>
<i>Internationalization</i>	<i>page 173</i>

Networking

The Solaris 2.5 operating environment includes the following networking features:

- Distributed file system (DFS), which centralizes the file system utilities
- Network information services plus (NIS+) including NFS
- Name service switch file

See *NIS+ and FNS Administration Guide* for more information on using these services.

NIS, NIS+

The Solaris 2.5 operating environment supports the network information service (NIS), the SunOS 4.x name service, and the network information services plus (NIS+), an enterprise-naming service of heterogenous distributed systems. See “NIS Support in SunOS Release 5.5” on page 135 for the nature of NIS support available in the Solaris 2.5 operating environment.

NIS+ provides a more detailed model for objects in the name space, improved security, and faster updates than NIS.

The NIS+ programmer interfaces are documented in section 3N of the *man Pages(3): Library Routines*.

nsswitch.conf File

The `nsswitch.conf` file is designed to simplify name service administration. Applications can use this file to select a name service. This information no longer needs to be hard-coded into the service itself. See the `nsswitch.conf(4)` man page for more information on the format of this file.

NIT

The Network Interface Tap (NIT) provided in the SunOS 4.x release is no longer required in the Solaris 2.5 operating environment. Now Ethernet drivers are real STREAMS drivers that can be opened and communicated with directly.

See `pfmod(7M)`, `bufmod(7M)`, and `dlpi(7P)`

The Solaris 2.5 Ethernet drivers and other data link drivers support the connectionless Data Link Provider Interface (DLPI) Version 2 specification.

Sockets

Sockets are supported in the Solaris 2.5 operating environment. Unlike the SunOS release 4.x, sockets are no longer implemented completely in the kernel. They are now a library, `libsocket`, implemented on STREAMS.

Internationalization

Developers concerned with internationalization will find that most of the changes in the Solaris 2.5 operating environment improve on previous internationalization features.

For complete information on internationalization support, see *Developer's Guide to Internationalization*.

Application developers concerned with internationalizing their programs should follow these guidelines:

- Call `setlocale(3C)` to set up the `LANG` environment variable
- Use standard code sets and follow 8-bit boundaries
- Use `strftime(3C)` to print the date and time
- Replace `strcmp(3)` with `strcoll(3C)` for user-visible collation
- Call `gettext(3C)` to retrieve translated strings from locale-specific message catalogs

Character Support

The Solaris 2.5 operating environment supports the extended UNIX code (EUC). This allows multibyte and multiple code sets on one system.

The SunOS release 4.x supported single byte representation of non-ASCII characters. The Solaris 2.5 operating environment supports multibyte representation. This support is needed for Asian language character sets, which contain thousands of characters.

The multibyte library, `libw`, is documented in section 3I of *man Pages(3): Library Routines* and provides the following features:

- Multibyte-to-wide character conversions
- Wide character standard I/O
- Wide character classification
- Wide character formatting

The Solaris 2.5 operating environment supports multibyte file names; however, login and machine names should be restricted to ASCII characters.

Message Catalogs

SunOS release 4.x support for message catalogs is enhanced in the Solaris 2.5 operating environment to enable the creation of message catalogs using multibyte characters.

Using message catalogs, an application can display messages at run time in the native language in which an application was run. These message catalogs must first be created for the native language specified by the language locale.

Commands

Most of the system commands in the Solaris 2.5 operating environment have been messaged. Many of these commands can pass through multibyte character representations. The increased number of messaged commands will ease localization efforts.

The `colldef(8)` command has been replaced with `colltbl(1M)`, which supports East Asian languages (level-4 support).

The `installtxt(1)` command has been replaced with `msgfmt(1)`. Use the new `xgettext(1)` command to extract messages.

The `montbl(1M)` command is now available to automatically generate monetary databases for `LC_MONETARY`.

Changes to `strftime(3C)` affect date and time formats. Shell programs that rely on the output format of the `date(1)` command will have to be updated to handle the new format.

Libraries

The `/usr/xpg2lib/libxpg2.a` archive library is no longer available. These routines have been included in `libc` or `libintl` (a new Solaris 2.5 library).

Table 17-1 shows the new location of these interfaces.

Table 17-1 xpg2lib Library Routine Locations

Routine	SunOS 4.x Location	Solaris 2.5 Location
bindtextdomain	/usr/xpg2lib/libxpg	/usr/lib/libintl
chroot	/usr/xpg2lib/libxpg	/usr/lib/libc
catgets	/usr/xpg2lib/libxpg	/usr/lib/libc
dgettext	/usr/xpg2lib/libxpg	/usr/lib/libintl
getcwd	/usr/xpg2lib/libxpg	/usr/lib/libc
getut	/usr/xpg2lib/libxpg	/usr/lib/libc
l3tol	/usr/xpg2lib/libxpg	/usr/lib/libc
logname	/usr/xpg2lib/libxpg	/usr/lib/libc
malloc	/usr/xpg2lib/libxpg	/usr/lib/libc
swab	/usr/xpg2lib/libxpg	/usr/lib/libc
langinfo	/usr/xpg2lib/libxpg	/usr/lib/libc
gettext	/usr/xpg2lib/libxpg	/usr/lib/libintl
sbrk	/usr/xpg2lib/libxpg	/usr/lib/libc
textdomain	/usr/xpg2lib/libxpg	/usr/lib/libintl

Programs that use these routines no longer need to pass `-lxpg2` to the C compiler (although some may need to specify `-lintl` now, and may need to include `libintl.h`; see Table 17-1 for these routines).

The `libintl` and `libw` libraries are new. The `libw` library provides wide character, multibyte support. These routines were included in the unbundled ALE product with previous releases, and are documented in section 3I of *man Pages(1): User Commands*.

The `catgetmsg(3C)` routine is no longer available.

The order of locale categories in the string returned by `setlocale(3C)` differs between the SunOS release 4.x and the Solaris 2.5 software. This string is normally used by a subsequent call to `setlocale(3C)`, and the order should not matter. Applications should not rely on a specific order of locale categories.

The operating system kernel and its interfaces have changed significantly since SunOS release 4.x. Binary compatibility is not provided for SunOS release 4.x device drivers. This chapter discusses changes in the Solaris 2.5 operating environment that affect kernel and system developers.

This chapter contains the following sections:

<i>System Configuration</i>	<i>page 177</i>
<i>Device Configuration: The Reconfiguration Boot</i>	<i>page 181</i>
<i>Device Naming From a Developer's Perspective</i>	<i>page 181</i>

System Configuration

Changes related to system configuration include the dynamically loaded kernel and kernel layout, the `config` and `boot` commands, and the `/etc/system` file.

Dynamically Loaded Kernel

Unlike previous SunOS releases, the kernel is now dynamically configured. The kernel now consists of a small static core and many dynamically loadable kernel modules. Drivers, file systems, STREAMS modules, and other modules are loaded automatically as needed, either at boot time or at runtime. When

these modules are no longer in use, they may be unloaded. Modules are kept in memory until that memory is needed. `modinfo(1M)` provides information about the modules currently loaded on a system.

The `modload(1M)` and `modunload(1M)` commands are still available in this release, but they perform differently. These commands have more limited usage in this release, and are no longer sufficient to correctly install a loadable driver onto the system. `modunload` is similar to the SunOS release 4.x command, but it includes the capability to unload all unloadable (and not busy) modules. Use `modunload` as follows.

```
# modunload -i 0
```

Kernel Layout

The contents of the kernel, which were formerly in a single file, `/vmunix`, are now contained in modules in a directory hierarchy. By default, the directory hierarchy is `/platform/'uname -i'/kernel`, `/kernel`, and `/usr/kernel`.

The directory search path for modules can be set by the `moddir` variable in the `/etc/system` file. See the `system(4)` man page in *man Pages(1M): System Administration Commands*. Typically, `/platform/'uname -i'/kernel/unix` is the first portion of the kernel to be loaded. See the `kernel(1M)` man page in *man Pages(1M): System Administration Commands*.

`config` *Command*

In SunOS release 4.x, the `config` command was used to generate system configuration files that allowed `/vmunix` to be relinked from object files. The need for this command has been removed by the following Solaris 2.5 features:

- Loadable modules
- The `/etc/system` file (see the `system(4)` man page)
- Device tree information from the OpenBoot PROM (OBP)
- The `driver.conf` files in `/kernel/drv` and `/usr/kernel/drv`

`/etc/system` *File*

System configuration information is now set in the `/etc/system` file. This file also modifies the kernel's treatment of loadable modules. The file contains commands of the form:

```
set parameter=value
```

For example, in SunOS release 4.x, MAXUSERS was set using `config(8)`. In the Solaris 2.5 operating environment, it is set in the `/etc/system` file with the following line.

```
set maxusers = number
```

Commands that affect loadable modules are of the form:

```
set module:variable=value
```

Changes made to the `/etc/system` file only take affect when you reboot your system (see the `system(4)` man pages).

`boot` *Command*

In this release, the following `boot` programs are available:

- `ufsboot` - To boot from a disk or a CD
- `inetboot` - To boot from across the network

When booting from a disk, the PROM assumes that the primary boot block resides in blocks 1 - 15 of the local disk. Use `installboot(1M)` to create the boot block:

```
# installboot /usr/platform/'uname' -i' /lib/fs/ufs/bootblk \  
/dev/rdisk/c0t3d0s0
```

The system firmware loads the primary bootstrap (the boot block) program into memory and runs it. The boot block is a UFS file system reader. It loads the secondary boot program (`/platform/'uname' -i' /ufsboot`) into memory.

ufsboot loads kernel/unix, then /kernel/unix uses ufsboot to load modules from the kernel directory hierarchy until it is able to mount the root file system.

During these operations, the boot block and ufsboot use the drivers provided by the firmware; neither ufsboot nor the boot block contains any driver code. The ufsboot code does not have to change to incorporate a new SBus card with a new disk type since ufsboot uses the SBus card PROM driver.

When booting over the network, the boot program performs as it did for a diskless boot in SunOS release 4.x. However, the boot program is now called inetboot and the client vfstab file entries are different. See *System Administration Guide, Volume I* for information on diskless booting.

Summary of Boot Differences

Table 18-1 summarizes the differences in the boot sequence between SunOS release 4.x and the Solaris 2.5 operating environment.

Table 18-1 Summary of Boot Differences

SunOS release 4.x	Solaris 2.5	Description
boot block	bootblk	Loads ufsboot from disk
boot program	ufsboot	Loads unix from disk
vmunix	unix	Bootable kernel image
boot.sun4c.sunos.4.1.1	inetboot	Mounts and copies unix from network
rc.boot, rc.single	/etc/rcS	Mounts /usr and checks file systems
rc.local	/etc/rc2, /etc/rc3, /etc/rc2.d, /etc/rc3.d	System configuration scripts
config	modload, /etc/system, add_drv, rem_drv	Customizes system kernel, loads, adds, and removes modules as needed
prom monitor, single user, multiuser	run states 0 – 6, and S	System run levels

Device Configuration: The Reconfiguration Boot

A reconfiguration boot tells the system to probe for all connected devices and build the names for them in `/devices` and `/dev`. A reconfiguration boot, performed when adding new hardware to the system is triggered by booting with the `-r` option, which follows.

```
ok boot -r
```

If another device of an existing type (with the driver already installed) is added, and you forget to do a reconfiguration boot, you can use the following commands to tell the system to recognize the new device.

```
# touch /reconfigure
# _INIT_RECONFIG=YES /etc/init.d/drvconfig
# _INIT_RECONFIG=YES /etc/init.d/devlinks
```

Device Naming From a Developer's Perspective

This section expands on the discussion in “Device Naming Conventions” on page 61, focusing on aspects of device naming that concern system and kernel developers.

/devices

The `/devices` tree represents the tree of devices recognized by the kernel. This tree is configured by the `drvconfig(1M)` program. `drvconfig` is normally run only when the system is booted with the `-r` flag. See “Device Configuration: The Reconfiguration Boot” on page 181. `drvconfig` configures `/devices` with information about devices (with drivers) that are connected and ready at boot time.

Entries are exported by device drivers calling `ddi_create_minor_node(9F)` when they have determined that a device exists.

Use the `add_drv(1M)` command to add a device to the system. If the driver was successfully added, `add_drv` will also run `drvconfig`.

/dev

In this release, `/dev` is managed by utility programs that create symbolic links to the real entries in `/devices`. The programs are:

- `disks(1M)`
- `tapes(1M)`
- `ports(1M)`
- `devlinks(1M)`

You can run a script to create the appropriate links from `/dev` to `/devices`. The `/dev` names have the advantage of being simpler and more familiar, while the `/devices` names are unique names for the hardware.

Device Driver Naming

Each device in the system is driven by a device driver. Device drivers manage many instances of a device. Devices are named in several ways:

- Physical names
- Logical names
- Instance names

Physical Names

Physical names are stored in `/devices`. They describe the hardware, and vary with the platform and configuration. For example:

```
/devices/vme/xdc@6d,ee80/xd@0,0:g
```

Physical names can be used to identify which piece of hardware is in use. For example:

```
xdc@6d,ee80
```

refers to the disk controller at address `0xee80` in VME A16, D32 space. See `vme(4)`, `driver.conf(4)` in *man Pages(1M): System Administration Commands*.

Logical Names

Logical names are stored in `/dev`. They attempt to abstract most of the nature of physical device names that are specific to the platform. Logical names might be appropriate for an `xd` device, such as:

```
/dev/dsk/c2d0s6 (controller 2, slave 0, slice 6 (4.x partition "g"))
```

or an `sd` device, such as:

```
/dev/dsk/c0t3d0s0 (controller 0, target 3, lun 0, slice 0 (4.x partition "a"))
```

The logical name conveys nothing about the type of controller. That is, it does not differentiate between SCSI and IPI; they are both just disks.

Disk Names

Disk names use the SVR4 convention of *slice* numbers 0–7 instead of the letters a–h used in SunOS release 4.x.

Disk names also use the SVR4 convention of `/dev/dsk/*` for block disk devices and `/dev/rdisk/*` for raw disks. For more information, see *System Administration Guide, Volume I*.

Instance Names

Instance names refer to the *n*th device in the system, for example, `sd20`.

Instance names are occasionally reported in driver error messages. You can determine the binding of an instance name to a physical name by looking at `dmesg(1M)` output, as in the following.

```
sd9 at esp2: target 1 lun 1
sd9 is /sbus@1,f8000000/esp@0,800000/sd@1,0
    <SUN0424 cyl 1151 alt 2 hd 9 sec 80>
```

Once the instance name has been assigned to a device, it remains bound to that device.

Instance numbers are encoded in a device's minor number. To keep instance numbers persistent across reboots, the system records them in the `/etc/path_to_inst` file. This file is read only at boot time, and is currently updated by the `add_drv(1M)` and `drvconfig(1M)` commands. See the `path_to_inst(4)` man page for more information.

This chapter discusses device driver issues such as changes to device driver interfaces, the `devinfo` command, porting considerations, STREAMS, and Solaris 2.5 driver architecture.

<i>Device Drivers and STREAMS Device Drivers</i>	<i>page 185</i>
<i>Device Driver Commands</i>	<i>page 193</i>

See the following guides for more information on the topics discussed in this chapter:

- *Writing Device Drivers*
- *STREAMS Programming Guide*
- *System Interfaces Guide*
- *System Administration Guide, Volume I*
- *System Administration Guide, Volume II*

Device Drivers and STREAMS Device Drivers

Some of the many changes to device drivers in the Solaris 2.5 operating environment include the new DDI/DKI routines, Solaris SPARC DDI-specific routines, new software properties, and loadable drivers. In addition, many previous device issues have become opaque to the driver. Including interrupts, DVMA, and memory mapping.

Device Driver Interfaces

In previous SunOS releases, a driver writer had to cope with changes in the device driver interfaces. Usually, there was a porting effort with each release of the operating system. In addition, the interfaces for each platform varied, so device drivers often required separate releases for each platform. Third party device driver releases often included complex scripts that would reconfigure and rebuild the operating system in order to integrate a device driver. It was costly to support and maintain device drivers.

The intention of the Solaris 2.5 SPARC DDI/DKI is to provide binary compatibility of device drivers across all supported platforms and for all future releases of the Solaris 2.5 operating environment on those platforms.

In the Solaris 2.5 operating environment, there is a new set of device driver interfaces. Unlike previous releases of SunOS systems (SunOS release 4.1.3 and earlier), the device driver interfaces in the Solaris 2.5 operating environment are formalized and are referred to as the *Solaris 2.5 SPARC DDI/DKI*.

The term *DDI/DKI* is derived from the original specification as supplied in the SVR4 release. It stands for *device driver interface/driver kernel interface*. The interfaces are divided into three groups:

- DDI/DKI
- DKI only
- DDI only

DDI/DKI

The *DDI/DKI interfaces* were standardized in SVR4, and are generic across all implementations of SVR4, regardless of the platform on which it is running.

DKI

The *DKI-only interfaces* are generic like the DDI/DKI interfaces and are supported in all SVR4 implementations. However, they are not guaranteed to be supported in future releases of System V.

DDI

The *DDI-only interfaces* are intended to be architecture-specific; for example, methods to access and control device and system-specific hardware (that is, I/O registers, DMA services, interrupts, and memory mapping). These interfaces are not guaranteed to work in other SVR4 implementations.

This group of features effectively lowers the cost of driver support and maintenance. These features, combined with the large number of SPARC platforms, are helpful to many new third party hardware developers.

By providing this level of binary compatibility, it is now possible for third-party hardware developers to “shrink-wrap” their DDI-compliant device drivers with their driver hardware. Installing a new driver package can now be entirely automated. The self-configuring kernel removes the necessity for recompiling the kernel to add or remove a driver. Thus, DDI-compliant device driver for Solaris 2.5 environments can be treated like any other consumer software product.

In the Solaris 2.5 DDI/DKI the DDI-only interfaces are generic to all SPARC-based systems that support the Solaris 2.5 DDI/DKI. Note that the interfaces that make up the Sun common SCSI architecture (SCSA), and the locking interfaces used to make the driver behave correctly in a multithreaded kernel are also considered DDI only interfaces in the Solaris 2.5 operating environment.

SCSA shields device drivers from details specific to the platform relating to host adapter implementations. With SCSA, a SCSI driver can run on all supported platforms.

A device driver that restricts itself to using only interfaces in the previous categories above is said to be *Solaris 2.5 DDI/DKI compliant*. A Solaris 2.5 DDI/DKI compliant device driver is commonly referred to as a *DDI-compliant* device driver.

Documentation

The man pages for the driver routines, structures, and support routines that comprise the DDI/DKI can be found in the following sections of *man Pages(1M): System Administration Commands*. See the `Intro(9)` man page for more information about these sections.

- Section 9E – Driver entry points
- Section 9F – Driver support functions
- Section 9S – Kernel structures

Also, see *Writing Device Drivers*. A Device Driver Developers Kit (DDK) is available separately.

devinfo *Command*

The Solaris 2.5 `devinfo` command performs a different function from the SunOS release 4.x version. The new `prtconf(1M)` command provides the information that the SunOS release 4.x `devinfo` command formerly displayed. The following examples show the output of each command.

```
4.1system% devinfo
Node 'SUNW,Sun 4/50', unit #0 (no driver)
  Node 'packages', unit #0 (no driver)
  Node 'openprom', unit #0 (no driver)
  Node 'zs', unit #0
  Node 'zs', unit #1
  Node 'audio', unit #0
  Node 'eeprom', unit #0 (no driver)
  Node 'counter-timer', unit #0 (no driver)
  Node 'memory-error', unit #0 (no driver)
  Node 'interrupt-enable', unit #0 (no driver)
  Node 'auxiliary-io', unit #0 (no driver)
  Node 'sbus', unit #0
    Node 'dma', unit #0
    Node 'esp', unit #0
      Node 'sr', unit #0
      Node 'sd', unit #0
    Node 'le', unit #0
    Node 'cgsix', unit #0
  Node 'memory', unit #0 (no driver)
  Node 'virtual-memory', unit #0 (no driver)
  Node 'fd', unit #0
  Node 'options', unit #0 (no driver)
```

```
5.3system% prtconf
System Configuration: Sun Microsystems sun4c
Memory size: 32 Megabytes
System Peripherals (Software Nodes):

SUNW,Sun 4_75
  packages (driver not attached)
    disk-label (driver not attached)
    deblocker (driver not attached)
    obp-tftp (driver not attached)
  openprom (driver not attached)
  zs, instance #0
  zs, instance #1
  audio (driver not attached)
  eeprom (driver not attached)
  counter-timer (driver not attached)
  memory-error (driver not attached)
  interrupt-enable (driver not attached)
  auxiliary-io (driver not attached)
  sbus, instance #0
    dma, instance #0
    esp, instance #0
      sd (driver not attached)
      st (driver not attached)
      sd, instance #0
      sd, instance #1 (driver not attached)
      sd, instance #2 (driver not attached)
      sd, instance #3
      sd, instance #4 (driver not attached)
      sd, instance #5 (driver not attached)
      sd, instance #6
    le, instance #0
    cgsix, instance #0
  memory (driver not attached)
  virtual-memory (driver not attached)
  fd (driver not attached)
  options, instance #0
  pseudo, instance #0
```

Porting Considerations

With the self-configuring kernel, Solaris 2.5 drivers will look more like SBUS drivers than other types. All drivers are loadable, and no kernel configuration

is required.

Under SunOS release 4.x, only one processor could be in the kernel at any one time. This was accomplished by using a *master lock* around the entire kernel. When a processor wanted to execute kernel code, it would acquire the lock (excluding other processors from running the code protected by the lock) and it would release the lock when it finished.

The Solaris 2.5 kernel is *multithreaded*. Instead of one master lock, there are many smaller locks that protect smaller regions of code. For example, there may be a kernel lock that protects access to a particular vnode, and one that protects an inode. Only one processor can be running code dealing with that vnode at a time, but another could be accessing an inode. This allows a greater amount of concurrency.

The multithreaded kernel will have a major impact on how you design the driver. The old model of using `splN/splr` pairs no longer works (on a uniprocessor or a multiprocessor system¹). Instead, you have a choice of MT-style locks to use. The most common of these for drivers will be mutual exclusion locks, *mutexes*, and condition variables (which are an approximate equivalent of `sleep()/wakeup()` synchronization).

The old notion that you *owned* the processor until you explicitly called `sleep()` is no longer true. Because of kernel preemption, the CPU is switched from thread to thread, so you *must* use the appropriate MT lock primitives to guard against concurrent access to device registers, shared data structures, and the like.

Writers of simple device drivers, which consist primarily of calls to kernel interface routines, will find that a large percentage of the driver code will change, but in straightforward ways. The writers of complex device drivers, such as a SCSI driver, which contain large amounts of device-specific handling code, will find that only a small percentage of the driver – the driver interfaces – changes. This driver interface can be a kernel to driver interface, a driver to kernel interface, or a driver to driver interface.

1. Strictly speaking, the `splN/splr` pair do work; however, it is ineffective on a multiprocessor system. That is, they will block interrupts, but the effect is useless in protecting data structures in a multiprocessor environment.

Before you determine how you will support the driver in the Solaris 2.5 operating environment, refamiliarize yourself with how the driver works. Determine what the SunOS release 4.x driver *did* (not the specific implementation, but general behavior). What interfaces did it export? What `ioctl`s did it provide? How did the hardware work and what peculiarities of the hardware did the driver support? Did the driver support multiple `open()` calls?

These changes will affect your driver and must be considered:

- The entry points to drivers are very different
- ANSI C requirements
 - `volatile` keyword
 - `const` keyword
 - Function prototype declarations
- Relocated or renamed header files (most, if not all, system header files are now in `/usr/include/sys`)
- Most structures have become opaque or are no longer needed. For example,
 - `struct user`
 - `struct proc`
 - `struct dev_info`

STREAMS

Some areas of change for STREAMS modules are transparent I/O controls, automatic pushing of modules on a stream, and new message types.

Transparent `ioctl()`s

In SunOS release 4.x, you had to know that a particular driver was a STREAMS driver before making `ioctl` requests.

For non-STREAMS drivers, you could do a direct `ioctl` request:

```
ioctl(fd, DRIVER_IOCTL, arg);
```

For a STREAMS driver, you had to set up a `strioc1` structure and then use:

```
ioctl(fd, I_STR, &strioc1);
```

There was no easy way to determine whether a driver was STREAMS-based. Now, unrecognized `ioctl`s to the stream head are passed on to the driver, eliminating the need to know whether a driver was STREAMS-based.

New message types were added in Solaris 2.5 software, in particular to support transparent `ioctl`s. There are now “copy in” and “copy out” messages to inform the stream head to transfer user data to and from the kernel.

For more information on writing STREAMS drivers, see the *STREAMS Programming Guide*.

`autopush` *Command*

Unlike SunOS release 4.x, a driver can no longer specify that certain STREAMS modules are pushed when the device is `open`(). The SunOS release 4.x `streamtab` structure provided this ability.

In the Solaris 2.5 operating environment, the system administrator and the `autopush(1M)` command specify when a STREAMS module is pushed. If required, `autopush` could be run at driver installation time.

See *STREAMS Programming Guide* for more information about pushing STREAMS modules.

Solaris 2.x Driver Architecture

To achieve binary compatibility across all currently supported hardware platforms, the DDI interfaces were carefully designed around architectural abstractions. The underlying abstraction, the `device tree`, is an extension of the `devinfo` tree in the original SPARCstation™ design. Each node in the device tree is described by a device information structure or “`dev_info` node.” The bottom-most nodes in the tree are termed *leaf nodes*. Most devices, such as disks and tape drives, framebuffers, I/O cards, and network interfaces, are examples of leaf devices that would be associated with leaf nodes. The associated device drivers are called *leaf drivers*.

The intermediate nodes in the tree are generally associated with buses (for example, SBus, SCSI, VME). These nodes are called *nexus nodes* and the drivers associated with them are called *nexus drivers*. Bus nexi are entities that are intended to encapsulate the architectural details associated with a particular element.

Currently, the Solaris 2.5 DDI/DKI supports only the writing of *leaf drivers* and one type of *nexus driver*, the SCSI host bus adapter driver.

The device tree structure creates a formal parent-child relationship between nodes. This parent-child relationship is the key to platform architecture independence.

When a leaf driver requires a service that is platform dependent (for example, a DMA mapping), the system transparently converts the request into a call to its parent to provide the service. The service providers are always nexus drivers; each nexus driver can in turn pass the request to its parent in order to provide the service. This approach allows leaf drivers to operate regardless of the platform architecture.

Device Driver Commands

The device driver commands are `add_drv`, `rem_drv`, `modload`, and `modunload`.

- `add_drv(1M)` – Informs the system that there is a newly installed device driver.
- `rem_drv(1M)` – Informs the system that the specified driver module is no longer valid.
- `modload(1M)` – Loads the specified loadable module into the running system.
- `modunload(1M)` – Unloads the specified loadable module from the running system.

Commands Reference Table



This appendix contains a user and system administration commands reference table that lists all SunOS release 4.x command interfaces, and shows their status in the Solaris 2.5 environment and the SunOS/BSD Source Compatibility Package.

Using the Reference Table

- If an interface is listed as “changed” (C), a brief description of differences between SunOS release 4.x command and the Solaris 2.5 command is provided.
- If an interface is listed as “the same” (S), the Solaris 2.5 interface supports all features of the SunOS release 4.x interface. In some cases the interface has been enhanced, but can be considered a complete superset of the SunOS release 4.x interface.
- If an interface has an “alternative” (A), check the Notes section for its replacement.
- If an interface is listed as “not available” (N), check the Notes section for information about its replacement. Replacement commands, when available, are also shown in the SunOS release 5.5 column.

Note – The directory structure in the SunOS release 5.5 release is different than the SunOS release 4.x structure; thus some commands behave the same, but have a different path name. For example, the SunOS release 4.x

`/usr/etc/newfs` command now resides in `/usr/sbin/newfs`, but the interface has not changed. This command, and others like it, are considered the same (S) according to this table's guidelines.

Commands that exist in both `/usr/bin` and `/usr/5bin` have two table entries, the first documents the `/usr/bin` command, and the second entry documents the `/usr/5bin` command.

For complete information on all Solaris 2.5 interfaces, see *man Pages(1): User Commands*.

Examples

Table A-1 through Table A-4 show table entries and are followed by an interpretation.

Table A-1 Example 1

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
<code>fasthalt(8)</code>	A	The <code>init 0</code> command provides similar capabilities.	S

The `fasthalt` command is not available in the Solaris 2.5 base release. This command is available if you install the SunOS/BSD Compatibility package on your system. The `init 0` command replaces `fasthalt`. If you use the compatibility package `fasthalt` command in scripts or applications, they will not work on other SVR4 systems. Compatibility package commands can be found in `/usr/ucb` on systems that have this package installed, and they are documented in section 1B of *man Pages(1): User Commands*, for example `fasthalt(1B)`.

Table A-2 Example 2

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
<code>cc(1V)</code>	N	The C compiler is only available with the C language unbundled tools.	C

The C compiler is not available in the SunOS release 5.5. A C compiler is available with the SunOS/BSD Compatibility package, but it requires the unbundled C compiler and does not provide the same interface and output as the SunOS release 4.x compiler. See “Compiler Option Differences” on page 140 for details.

Table A-3 Example 3

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
date(1V) -- SysV	S		N
date(1V)	C	The format used when setting the date is slightly different in SunOS release 5.5. See the date(1) man page for more information.	N

SunOS release 4.x had two date commands: /usr/5bin/date (compared in the SysV entry) and /usr/bin/date (compared in the second entry). The /usr/5bin/date command is identical to the SunOS release 5.5 command. If you had /usr/5bin in your path before /usr/bin, you will not notice any difference in this command in the SunOS release 5.5. If you are accustomed to using SunOS release 4.x /usr/bin/date command, you should look at the SunOS release 5.5 date(1) man page before attempting to set the date on your system.

Table A-4 Example 4

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
rev(1)	N		N

The SunOS 4.2 rev command is not available in the SunOS release 5.5 or the BSD release. There is no replacement command available.

The Commands Reference Table

Table A-5 lists all SunOS release 4.x command interfaces, and shows their status in the Solaris 2.5 environment and in the SunOS/BSD Source Compatibility Package.

Table A-5 Commands Reference Table

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
C2conv(8)	N	See your system vendor for information on this product.	N
C2unconv(8)	N	See your system vendor for information on this product.	N
Mail(1)	A	The mailx(1) command provides similar capabilities.	N
ac(8)	A	The System Accounting Resource package (SAR) provides most of the accounting capabilities available in ac.	N
acctcms(8)	S		N
acctcom(8)	S		N
acctcon1(8)	S		N
acctcon2(8)	S		N
acctdisk(8)	S		N
acctdusg(8)	S		N
acctmerg(8)	S		N
accton(8)	S		N
acctprc1(8)	S		N
acctprc2(8)	S		N
acctwtmp(8)	S		N
adb(1)	S		N
adbgen(8)	S		N
add_client(8)	N	admintool(1M)	N
add_services(8)	A	The swmtool(1M) command provides similar capabilities.	N
addbib(1)	S		N
adjacentscreens(1)	OpenWindows	The OpenWindows environment has two methods for providing multiple displays: Start two servers on a given machine, each server controlling its specific display. Start one server with two displays, using the openwin -dev option.	N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
admin(1)	C	The following SunOS release 4.x options are not available in the SunOS release 5.5 system software: -lrelease[,release . . .] Lock indicated release against deltas.	N
adv(8)	N	RFS does not exist. This capability is still accessible via the -f flag.	N
aedplot(1G)	N		S
align_equals(1)	OpenWindows	The OpenWindows text menu indent command provides similar capabilities.	N
analyze(8)	A	Use adb(1) on core files to analyze crashes.	N
apropos(1)	C	The SunOS release 4.x command used the <code>whatis</code> database in SunOS release 5.5, this database is called <code>windex</code> , and the format is slightly different.	N
ar(1V)	S		N
ar(1V) -- SysV	C		N
arch(1)	S	Similar capabilities are available with <code>uname(1)</code>	S
arp(8C)	S		N
as(1)	C	The following SunOS release 4.x options are not available in the SunOS release 5.5 command: -d2, -h, -j, -J, -k, -O[n].	N
at(1)	S	The <code>at</code> , <code>atq</code> , and <code>atrm</code> commands in SunOS release 5.5 systems behave slightly differently than they do in SunOS release 4.x systems. Security for non-privileged users is more restricted on SunOS release 5.5 systems. Non-privileged users cannot display the jobs of any other user.	N
atoplot(1G)	N		S

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
atq(1)	C	The at, atq, and atrm commands in SunOS release 5.5 systems behave slightly differently than they do in SunOS release 4.x systems. In the SunOS release 4.x command, if no user name is specified, the entire queue is displayed. In SunOS release 5.5 system software, the entire queue is displayed only if the invoker is a privileged user; otherwise, only the jobs belonging to the invoker are displayed. A non-privileged user cannot list the jobs of another user. Security for non-privileged users is more restricted on SunOS release 5.5 systems.	N
atrm(1)	C	The at, atq, and atrm commands in SunOS release 5.5 systems behave slightly differently than they do in SunOS release 4.x systems. The SunOS release 4.x '-' flag has been renamed to -a in the SunOS release 5.5 command. Security for non-privileged users is more restricted on SunOS release 5.5 systems.	N
audit(8)	C	-d or -u options are not available. This command is available only if the Basic Security Module (BSM) has been enabled.	N
audit_warn(8)	S		N
auditd(8)	S		N
automount(8)	C	The following SunOS release 4.x option is not available in the SunOS release 5.5 command: -m Suppress initialization of directory-map pairs. The auto.master and auto.home files are renamed auto_master and auto_home in SunOS release 5.5. The default home directory path is /export/home/ <i>username</i> .	N
awk(1)	S		N
banner(1V) -- SysV	S		N
bar(1)	tar, cpio	The tar(1) command can replace bar for most uses. You can use cpio -iH bar to restore existing SunOS release 4.x bar backups. It is no longer possible to create bar format files in this release.	N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
basename(1)	S	The SunOS release 5.5 and SunOS/BSD Compatibility versions are both compatible to the SunOS release 4.x version, but they differ in how they parse arguments: the SunOS release 5.5 version will not accept more than two arguments, the SunOS/BSD Compatibility version ignores all arguments after the second.	S
batch(1)	S	By default, the SunOS release 5.5 batch job <i>queuename</i> is not specified, jobs were always queued on queue b with the SunOS release 4.x command.	N
bc(1)	S		N
bgplot(1G)	N		S
biff(1)	chmod	fiff n: % chmod u+x 'tty' biff y: % chmod u-x 'tty'	S
bin-mail(1)	S	Same as the SunOS release 5.5 mail(1) command.	N
biod(8)	N		N
boot(8S)	C	See the boot(1M) man page for more information.	N
bootparamd(8)	S		N
cal(1)	S		N
calendar(1)	S		N
cancel(1)	S		N
capitalize(1)	OpenWindows	An OpenWindows version of this command is available with the OpenWindows text editor.	N
captainfo(8V) -- SysV	S		N
cat(1V) -- SysV	S		N
cat(1V)	S	The SunOS release 5.5 cat command requires the -v option with the -t and -e options. The SunOS release 5.5 command displays FORMFEED characters with the -t option, instead of the -v option as with the SunOS release 4.x command.	N
catman(8)	S		N
cb(1)	S		N
cc(1V) -- SysV	N		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
cc(1V)	N	The C compiler is only available with the C language unbundled tools.	C
cd(1)	S		N
cdc(1)	C	The two versions differ in how they treat an unreadable s.file. The SunOS release 4.x command prints an error; the SunOS release 5.5 command silently ignores the error.	N
cflow(1V) -- SysV	N	The cflow command is now available as an unbundled product.	N
cflow(1V)	N	The cflow command is now available as an unbundled product.	N
chargefee(8)	S		
checkeq(1)	S		N
checknr(1)	S		N
chfn(1)	N		N
chgrp(1)	C	The default behavior of symbolic links has changed from SunOS release 4.x to SunOS release 5.5 system software. In SunOS release 4.x system software, chgrp changed ownership of the symbolic itself; in SunOS release 5.5 system software, chgrp follows the link. To change ownership of the symbolic link in SunOS release 5.5 system software, use the -h option.	N
chkey(1)	S		N
chmod(1V) -- SysV	C	The SunOS release 5.5 -R option changes the mode of the target when symbolic links are encountered.	N
chmod(1V)	S	The SunOS release 5.5 -R option changes the mode of the target when symbolic links are encountered. The SunOS release 5.5 command supports two additional permissions: 'l' and 't'.	N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
chown(8)	C	The default behavior of symbolic links has changed. SunOS release 4.x chown changed ownership of the symbolic link. SunOS release 5.5 chown follows the link. To change the ownership of the link, use chown -h. The SunOS release 5.5 chown command does not allow changing the group ID of a file.	S
chroot(8)	S		N
chrtbl(8)	S		N
chsh(1)	N		N
ckpacct(8)	S		N
clear(1)	S		N
clear_colormap(1)	N		N
clear_functions(1)	S		N
click(1)	N		N
clock(1)	OpenWindows	An OpenWindows command is available in /usr/demo/clock. See the clock(1) man page for information.	N
clri(8)	S		N
cmdtool(1)	OpenWindows	This command is replaced by the OpenWindows Command Tool.	N
cmp(1)	S		N
col(1V) -- SysV	S		N
col(1V)	C		N
colcrt(1)	N		N
colldef(8)	colltbl	The colltbl(1M) command provides similar capabilities.	N
coloredit(1)	OpenWindows	The function of this command is now handled by the OpenWindows property window.	N
colrm(1)	N		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
comb(1)	C	The two versions differ in how they treat an unreadable s.file. The SunOS release 4.x command prints an error, but the SunOS release 5.5 command silently ignores the error.	N
comm(1)	S		N
compress(1)	S		N
config(8)	N		N
cp(1)	C	The -R option is replaced by the -r option in the SunOS release 5.5 command.	N
cpio(1)	S		N
cpp(1)	S		N
crash(8)	C	The default name list used in SunOS release 4.x is /vmunix, but it is /kernel/unix in SunOS release 5.5	N
cron(8)	S		N
crontab(1)	S		N
crtplot(1G)	N		S
crypt(1)	S		N
cs(1)	S		N
csplit(1V) -- SysV	S		N
ctags(1)	S		N
ctrace(1V) -- SysV	N	The following SunOS release 4.x option is not available in the SunOS release 5.5 command: -b Use only basic functions to trace code. This option is needed for running under an operating system that does not have the signal(), fflush(), longjmp() or setjmp() functions available. The syntax of the -r option differs between SunOS release 4.x and SunOS release 5.5 system software. The 4.1 format is -rf; in SunOS release 5.5, it is -r f. ctrace is available as an unbundled product.	N
cu(1C)	S		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
cut(1V) -- SysV	S		N
cxref(1V) -- SysV	S		N
cxref(1V)	N	cxref is available as an unbundled product.	N
date(1V) -- SysV	S		N
date(1V)	C	The format used when setting the date is slightly different in SunOS release 5.5. See the date(1) man page for more information.	N
dbconfig(8)	S		N
dbx(1)	N	Available with the unbundled SPARCworks product.	N
dbxtool(1)	N	Available with the unbundled SPARCworks product as the command debugger.	N
dc(1)	S		N
dcheck(8)	A	Use the fsck(1M) command for normal consistency checking. The ncheck(1M) command replaces the function of dcheck - i numbers.	N
dd(1)	C	In the SunOS release 4.x command, the size used for the size suffix w (words) is in units of 4 bytes, while in SunOS release 5.5 system software, w is in units of 2 bytes. k, b, or w may be used as a suffix to specify multiplication by 1024, 512, or 2, respectively. The unblock and block conversion options are new with SunOS release 5.5.	N
defaults_from_input(1)	OpenWindows	The function of this command is now handled by the OpenWindows property window.	N
defaults_merge(1)	S		N
defaults_to_indentpro(1)	OpenWindows	The function of this command is now handled by the OpenWindows property window.	N
defaults_to_mailrc(1)	OpenWindows	The function of this command is now handled by the OpenWindows property window.	N
defaultsedit(1)	OpenWindows	The function of this command is now handled by the OpenWindows property window.	N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
delta(1)	C	If a directory is specified as the argument, all files in the directory are processed. In SunOS release 4.x, an error is produced if a file in a directory generates an error. Such files are silently ignored by the SunOS release 5.5 command.	N
deroff(1)	S		N
des(1)	S		N
devinfo(8S)	C	The <code>prtconf(1M)</code> command provides similar capabilities.	N
devnm(8)	C	The output format between SunOS release 4.x and SunOS release 5.5 system software is quite different. In SunOS release 4.x system software, the name argument is optional. In the SunOS release 5.5 system software, it is required.	N
df(1V) -- SysV	C		N
df(1V)	C	The SunOS release 4.x version of this command provides a different output format containing somewhat different output than the SunOS release 5.5 <code>df</code> command. The SunOS release 5.5 <code>-k</code> option provides output formats similar to those in the SunOS release 4.x command. The SunOS release 4.x <code>df -t filesystem</code> type reports on files of the specified type, whereas the SunOS release 5.5 <code>df -t</code> command prints full listings with totals. You can use <code>df -l</code> to see local filesystems.	S
diff(1)	C	The behavior of several flags differs between the two versions. In SunOS release 4.x system software, the <code>-c</code> option takes an optional argument for the number of lines to display for each difference. If no argument is given, the default is 3 lines. In the SunOS release 5.5 command, a space is required between the <code>-s</code> option and its argument.	N
diff3(1V) -- SysV	S		N
diff3(1V)	S		N
diffmk(1)	S		N
dircmp(1V) -- SysV	S		N
dirname(1V) -- SysV	S		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
dis(1)	C	The following SunOS release 4.x option is not available in the SunOS release 5.5 command: -da <i>sec</i> Disassemble <i>sec</i> as data, printing the actual address of the data. Use the SunOS release 5.5 -D <i>sec</i> option to do the same thing.	N
diskusg(8)	A	The acctdusg (1M) command provides similar capabilities.	N
dkctl(8)	N		N
dkinfo(8)	prvtoc	The prvtoc(1M) command provides similar capabilities.	N
dmesg(8)	S		N
dname(8)	N	RFS is not available.	N
dodisk(8)	S		N
domainname(1)	S		N
dorfs(8)	N	RFS is not available.	N
dos2unix(1)	S		N
du(1V) -- SysV	S		N
du(1V)	C	The SunOS release 4.x command reports the disk usage in kilobytes while the SunOS release 5.5 du command reports disk usage in 512-byte blocks. The -k option can be used to report usage in kilobytes.	S
dumbplot(1G)	N		S
dump(8)	A	The ufsdump command provides similar capabilities. The following SunOS release 4.x options are not in the SunOS release 5.5 command: -a <i>archive-file</i> The SunOS release 5.5 -a option dumps the archive header of each member of an archive. -D Specify diskette as the dump media. The SunOS release 5.5 -D option dumps debugging information. -v Verify against the file system being dumped. The SunOS release 5.5 -v option dumps information in symbolic, rather than numeric, representation.	N

≡ A

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
dumpfs(8)	A	The <code>fstyp -F ufs -v</code> command provides similar capabilities.	N
dumpkeys(1)	S		N
e(1)	A	The <code>ex(1)</code> command provides similar capabilities.	S
echo(1V) -- SysV	S		N
echo(1V)	C	The <code>-n</code> option suppressed new-line printing under SunOS release 4.x. Use a <code>\c</code> for SunOS release 5.5.	S
ed(1)	S		N
edit(1)	S		N
edquota(8)	S		N
EEPROM(8S)	S		N
egrep(1V)	S		N
eject(1)	S		N
enroll(1)	N		N
env(1)	S		N
eqn(1)	S		N
error(1)	S		N
etherd(8C)	A	The <code>snoop(1M)</code> command provides similar capabilities.	N
etherfind(8C)	A	The <code>snoop(1M)</code> command provides similar capabilities.	N
ex(1)	S		N
expand(1)	S		N
exportfs(8)	A	The <code>share(1M)</code> command provides similar capabilities.	N
expr(1V) -- SysV	S		N
expr(1V)	C		S
extract_files(8)	A	The <code>pkgadd(1M)</code> command provides similar capabilities.	N
extract_patch(8)	A	The <code>pkgadd(1M)</code> command provides similar capabilities.	N
extract_unbundled(8)	A	The <code>swmtool(1M)</code> command provides similar capabilities.	N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
false(1)	S		N
fastboot(8)	A	The <code>init 6</code> command provides similar capabilities.	S
fasthalt(8)	A	The <code>init 0</code> command provides similar capabilities.	S
fdformat(1)	S		N
fgrep(1V)	S		N
file(1)	C	The following SunOS release 4.x option is not in the SunOS release 5.5 command: -L If a file is a symbolic link, test the file referenced by the link rather than the link itself.	S
find(1)	C	The following SunOS release 4.x option is not available in the SunOS release 5.5 command: -n <i>cpio-device</i> Write the current file on device in <code>cpio -c</code> format.	N
finger(1)	S		N
fingerd(8)	S		N
fmt(1)	C		N
fmt_mail(1)	N		N
fold(1)	S		N
fontedit(1)	N		N
foption(1)	N		N
format(8S)	S		N
fpa_download(8)	N		N
fparel(8)	N		N
fpaversion(8)	N		N
fpurel(8)	N		N
fpuversion4(8)	A	This information is available from <code>psrinfo -v</code>	N
from(1)	N		S

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
fsck(8)	C	The SunOS release 4.x <code>fsck</code> command differs significantly from the SunOS release 5.5 command. With the SunOS release 5.5 command, you specify most options after you specify the file system type. <code>fsck -m</code> does a quick file-system check. The <code>-w</code> option is not available. New options include <code>-f</code> , <code>-v</code> , and <code>-o</code> .	N
fsck-cdrom(8)	N		N
fsirand(8)	S		S
ftp(1C)	S		N
ftpd(8C)	S		N
fumount(8)	S	RFS is no longer available	N
fusage(8)	S	RFS is no longer available	N
fuser(8)	S		N
fwtmp(8)	S		N
gcore(1)	S		N
generic_args(1)	N		N
get(1)	C	The SunOS release 5.5 command generates only ASCII files; there is no such restriction in SunOS release 4.x system software. If a directory is specified and the files inside the directory cannot be obtained successfully, the SunOS release 4.x command reports an error; the SunOS release 5.5 command ignores them silently.	N
get_alarm(1)	N		N
get_selection(1)	A	The <code>xv_get_sel(1)</code> command provides similar capabilities.	N
getopt(1V) -- SysV	S		N
getoptcvt(1)	S		N
getopts(1)	S		N
gettable(8C)	S		N
getty(8)	S		N
gfxtool(1)	N		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
gigiplot(1G)	N		S
glob(1)	S		N
goto(1)	S		N
gpconfig(8)	N		N
gprof(1G)	S		N
graph(1G)	S		N
grep(1V)	S		N
grep(1V) -- SysV	C	The following option has changed: -w Search for the regular expression as a word as if surrounded by \< and \>.	N
groups(1)	S		S
grpck(8V)	S		N
gxtest(8S)	N		N
halt(8)	S		N
hashcheck(1)	S		N
hashmake(1)	S		N
hashstat(1)	S		N
head(1)	S		N
help(1)	S		N
help_open(1)	S		N
hostid(1)	S		S
hostname(1)	S		S
hostrfs(8)	N	RFS is not available.	N
hp7221plot(1G)	N		S
hpplot(1G)	N		S
htable(8)	S		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
i386(1)	S		N
iAPX286(1)	S		N
icheck(8)	A	<code>fsdb()</code> is an alternate command.	N
iconedit(1)	OpenWindows	This command is replaced by the OpenWindows Icon Edit tool.	N
id(1)			
id(1V) -- SysV	S		N
idload(8)	N	RFS is not available.	N
ifconfig(8C)	S		N
imentest(8C)	N		N
implot(1G)	N		Y
in.comsat(8C)	S		N
in.fingerd(8C)	S		N
in.ftpd(8C)	S		N
in.named(8C)	S		N
in.rexecd(8C)	S		N
in.rlogind(8C)	S		N
in.routed(8C)	S		N
in.rshd(8C)	C	The port range differs between the SunOS release 4.x and SunOS release 5.5 commands. In SunOS release 4.x system software, the range is 512-1023; in SunOS release 5.5 system software, it is 0-1023.	N
in.rwhod(8C)	S		N
in.talkd(8C)	S		N
in.telnetd(8C)	S		N
in.tftpd(8C)	S		N
in.tnamed(8C)	S		N
in.uucpd(8C)	S		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
indent(1)	N	This command is now available as an unbundled product.	N
indentpro_to_defaults(1)	OpenWindows	The function of this command is now handled by the OpenWindows property sheets.	N
indxbib(1)	S		N
inetd(8C)	S		N
infocmp(8V) -- SysV	C		N
infocmp(8V)	C	The syntax of the <code>-s</code> option differs between SunOS release 4.x and SunOS release 5.5 system software. In the SunOS release 5.5 command, there must be a space between <code>-s</code> and its argument. In the SunOS release 4.x command, the space is optional.	N
init(8)	C	The SunOS release 5.5 command is very different from the SunOS release 4.x command. See the <code>init(1M)</code> man page for more information.	N
inline(1)	N	This command is now available as an unbundled product.	N
input_from_defaults(1)	N		N
insert_brackets(1)	A	An OpenWindows command with the same name is available with the OpenWindows Text Editor.	N
install(1)	C	The functions of the <code>-c</code> , <code>-o</code> , and <code>-s</code> options are different between the SunOS release 4.x and SunOS release 5.5 commands.	S
installboot(8S)	C	The pathnames and syntax have changed.	N
installtxt(8)	A	The <code>msgfmt(1)</code> command provides similar capabilities.	N
intr(8)	N		N
iostat(8)	S	New options: <code>-x</code> Provide disk statistics <code>-c</code> Report the percentage of time the system has spent in user mode, system mode, and idle.	N
ipallocald(8C)	N		N
ipcrm(1)	S		N
ipcs(1)	S		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
join(1)	C	In the SunOS release 4.x command, the <code>-a</code> option takes an argument whose value can be 1, 2, or 3. In SunOS release 5.5 system software, this value can only be 1 or 2. In the SunOS release 4.x command, the argument to <code>-j</code> can only be 1 or 2; there is no such restriction in the SunOS release 5.5 command.	N
kadb(8S)	S		N
keyenvoy(8C)	N		N
keylogin(1)	S		N
keylogout(1)	S		N
keyserv(8C)	S		N
kgmon(8)	S		N
kill(1)	S		N
labelit(8)	S		N
last(1)	S		N
lastcomm(1)	S		N
lastlogin(8)	S		N
ld(1)	C	There are many differences between the SunOS release 4.x <code>ld</code> command and the SunOS release 5.5 command. The following SunOS release 4.x options are not available with SunOS release 5.5: <code>-align</code> , <code>-A</code> , <code>-B</code> , <code>-D</code> , <code>-M</code> , <code>-n</code> , <code>-t</code> , <code>-T</code> , <code>-Tdata</code> , <code>-x</code> , <code>-X</code> , <code>-y</code> and <code>-z</code> . The <code>-assert</code> option has been replaced in SunOS release 5.5 by the <code>-z</code> option. The <code>-d</code> , <code>-dc</code> , <code>-dp</code> options are the default in SunOS release 5.5 system software. To turn off these options use <code>-b</code> .	S
ldconfig(8)	N		N
ldd(1)	S		N
leave(1)	N	The <code>cron(1M)</code> and <code>at(1)</code> commands provide similar capabilities.	N
lex(1)	C	The following SunOS release 4.x option is not available in the SunOS release 5.5 command: <code>-f</code> Compile faster by not packing resulting tables. This option is limited to small programs.	N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
line(1)	S		N
link(8V)	S		N
lint(1V) -- SysV	N		N
lint(1V)	N	Available with unbundled SPARCworks product.	S
listen(8)	S		N
ln(1V)	C	The SunOS release 4.x <code>ln</code> command never removes the target if it already exists. The SunOS release 5.5 <code>ln</code> command removes the target, given the proper permissions. The SunOS release 4.x <code>-f</code> option forces a hard link to a directory.	S
ln(1V) -- SysV	C	In SunOS release 4.x <code>/usr/5bin/ln</code> , the <code>-f</code> option forces files to be linked without displaying permissions, asking questions, or reporting errors. The <code>/usr/5bin/ln -F</code> option to force a hard link to a directory is not available in SunOS release 5.5 system software.	N
loadkeys(1)	S		N
lockd(8C)	S		N
lockscreen(1)	A	This command is available as the OpenWindows tool <code>xlock(1)</code> . The capabilities of the <code>lockscreen</code> command remains the same in <code>xlock</code> , although the foreground pattern differs.	N
logger(1)	N		S
login(1)	S		N
logname(1)	S		N
look(1)	S		N
lookbib(1)	S		N
lorder(1)	S		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
lp(1)	S		N
lpc(8)	A	The lpadmin(1M) command provides similar capabilities.	S
lpd(8)	A	The lpadmin(1M) command provides similar capabilities.	S
lpq(1)	A	The lpstat(1) command provides similar capabilities.	S
lpr(1)	A	The lp(1) command provides similar capabilities.	S
lprm(1)	A	The cancel(1) command provides similar capabilities.	S
lpstat(1)	S		N
lptest(1)	N		S
ls(1V) -- SysV	C		N
ls(1V)	S		S
lsw(1)	N		N
m4(1V)	C	Some small syntactic incompatibilities over expression evaluation.	N
m4(1V) -- SysV	S		N
m68k(1)	S		N
mach(1)	S		S
mail(1) -- UCB	mailx		S
mail(1)	C	Now in /usr/bin/mail, was in /usr/ucb/mail in SunOS release 4.x. This entry refers to the mail command installed under /usr/bin/mail. The SunOS release 4.x mail is compatible with the SunOS release 5.5 command except for the following: -i The -i (ignore interrupts) option is not available with SunOS release 5.5. postmark line In the SunOS release 4.x command, the postmark line is preceded by a '>'; this is not required by the SunOS release 5.5 command. mail forwarding	N
mailrc_to_defaults(1)	OpenWindows	The function of this command is now handled by the OpenWindows property window.	N
mailstats(8)	S		N
mailtool(1)	OpenWindows	This command is available as the OpenWindows Mail Tool.	N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
make(1)	S	SVR4 & SVID make is available in <code>/usr/ccs/lib/svr4.make</code>	N
makedbm(8)	C	The SunOS release 5.5 interface for this command is compatible with the SunOS release 4.x interface. The SunOS release 5.5 version uses <code>/usr/lib/ndbm</code> rather than <code>/usr/lib/dbm</code> as the SunOS release 4.x version does.	N
makedev(8)	N		N
makekey(8)	S		N
man(1)	C	The organization of the on-line man pages has changed. Refer to <code>intro(1)</code> for a description of all sections. The <code>man</code> command now allows you to specify a default order of directories for <code>man</code> to search. Two new options make it easier to find man pages: -a to display all man pages matching <i>title</i> in the order found; and -l to list all man pages matching <i>title</i> . Also, the <code>-s</code> option replaces the <i>section number</i> argument.	N
mc68010(8)	S		N
mc68020(8)	S		N
mc68881version(8)	N		N
mconnect(8)	S		N
mesg(1)	S		N
mkdir(1)	S		N
mkfile(8)	S		N
mkfs(8)	C	The interface differs significantly between the two versions. The SunOS release 5.5 command provides for different file system types.	N
mknod(8)	S		N
mkproto(8)	C		N
mkstr(1)	N		S
modload(8)	C	Modules are usually automatically loaded using <code>modload</code> .	N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
modstat(8)	modinfo	The modinfo(1M) command provides similar capabilities.	N
modunload(8)	C	Modules are usually automatically unloaded.	N
monacct(8)	S		N
more(1)	S		N
mount(8)	C	The interface differs significantly between the two versions. In the SunOS release 5.5 version, most options must be specified after the file system type has been specified (unless the file system is entered in <code>/etc/vfstab</code>).	N
mount_tfs(8)	N		N
mountd(8C)	S		N
mt(1)	S		N
mv(1)	S		N
named(8C)	C	The name daemon is renamed to <code>in.named</code> .	N
nawk(1)	S		N
ncheck(8)	C	Modified to allow specification of different file system types.	N
ndbootd(8C)	N		N
neqn(1)	S		N
netstat(8C)	S		N
newaliases(8)	S		N
newfs(8)	S	Moved from <code>/usr/etc/newfs</code> to <code>/usr/sbin/newfs</code> .	N
newgrp(1)	S		N
newkey(8)	S		N
nfsd(8)	S		N
nfsstat(8C)	S		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
nice(1)	C	There are two versions of <code>nice</code> in SunOS release 4.x system software, one built into the <code>cs</code> h and one installed under <code>/usr/bin</code> . The default process priority for the command built into <code>cs</code> h is 4, and the default value for <code>/usr/bin/nice</code> is 10. The SunOS release 5.5 command defaults to 10. The SunOS release 4.x command that is built into the <code>cs</code> h uses a slightly different syntax than the SunOS release 4.x command found in <code>/usr/bin</code> , in that the additional <code>+</code> option (<code>nice +n</code>) sets the <code>nice</code> value to <code>n</code> rather than incrementing it by <code>n</code> .	N
nl(1V) -- SysV	S		N
nlsadmin(8)	C	The function of the <code>-l</code> option differs between the versions. In SunOS release 4.x, changing <code>addr</code> does not take effect until the next time the listener for that network is started. In SunOS release 5.5, it happens immediately. In SunOS release 4.x, <code>addr</code> can be specified in hexadecimal notation while in SunOS release 5.5 it cannot. The SunOS release 4.x <code>-m</code> option is not available in the SunOS release 5.5 version. This option is used to add a new service to the list of services available through the indicated listener.	N
nm(1)	C	The following SunOS release 4.x options are not available with the SunOS release 5.5 version: <code>-g</code> , <code>-p</code> , <code>-s</code> , and <code>-a</code> . The SunOS release 4.x and SunOS release 5.5 versions of the <code>-n</code> , <code>-o</code> and <code>-r</code> options differ.	N
nohup(1V)	C		N
nohup(1V) -- SysV	S		N
nroff(1)	S		N
nslookup(8C)	S		N
nsquery(8)	S		N
nulladm(8)	S		N
od(1V)	S		N
od(1V) -- SysV	S		N
old-analyze(8)	N		N

≡ A

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
old-ccat(1)	N		N
old-clocktool(1)	N		N
old-compact(1)	N		N
old-eyacc(1)	N		N
old-filemerge(1)	N		N
old-make(1)	N		N
old-perfmon(1)	N		N
old-prmail(1)	N		N
old-pti(1)	N		N
old-setkeys(1)	N		N
old-sun3cvt(1)	N		N
old-syslog(1)	N		N
old-uncompact(1)	N		N
old-vc(1)	N		N
on(1C)	S		N
overview(1)	N		N
pac(8)	N		N
pack(1V)	S		N
pack(1V) -- SysV	S	With the SunOS release 4.x /usr/5bin/pack command, file names are restricted to 12 characters. In SunOS release 5.5 system software, they are restricted to {NAME_MAX} - 2. The SunOS release 5.5 pack and unpack commands are compatible with the SunOS release 4.x commands.	N
page(1)	S		N
pagesize(1)	S		S

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
passwd(1)	C	The <code>-F filename</code> option is not available. The <code>-f</code> and <code>-s</code> options have different meanings. The <code>-f</code> option forces the user to change the password at the next login. The <code>-s</code> option displays the password attributes for the user's login name.	N
paste(1V) -- SysV	S		N
pax(1V)	C		N
paxcpio(1V)	A	The <code>cpio(1)</code> and <code>pax(1)</code> commands provides similar capabilities.	N
pcat(1V) -- SysV	S		N
pdp11(1)	S		N
perfmeter(1)	OpenWindows	This command is available in SunOS release 5.5 as the OpenWindows Performance Meter tool.	N
pg(1V) -- SysV	S		N
ping(8C)	S		N
plot(1G)	N		S
plottoa(1G)	N		S
portmap(8C)	A	The <code>rpcbind(1M)</code> daemon provides similar capabilities.	N
pr(1V)	C		N
pr(1V) -- SysV	S		N
praudit(8)	S		N
prctmp(8)	S		N
prdaily(8)	S		N
printenv(1)	A	The <code>env(1)</code> command provides similar capabilities.	S

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
prof(1)	C	The SunOS release 4.x <code>-v</code> option is not available with SunOS release 5.5 system software. This option suppresses all printing and produce a graphic version of the profile on the standard output for display by the <code>plot(1)</code> filters. The SunOS release 4.x <code>-a</code> option requests that all symbols be reported, in the SunOS release 5.5 command, just external symbols are reported.	N
prs(1)	C	The versions differ in how they treat an unreadable <code>s.file</code> . The SunOS release 4.x command prints an error and continues if it encounters an unreadable <code>s.file</code> . SunOS release 5.5 silently ignores the error.	N
prt(1)	S		N
prtacct(8)	S		N
ps(1)	C	The following SunOS release 4.x options are not available with SunOS release 5.5 system software: <code>C</code> , <code>k</code> , <code>n</code> , <code>r</code> , <code>S</code> , <code>U</code> , <code>v</code> , <code>w</code> , and <code>x</code> . The following options have different meanings in the two versions: <code>-c</code> In the SunOS release 4.x command, this option displays the command name. In the SunOS release 5.5 command, it prints information in a format that reflects the new process scheduler design.	S
pstat(8)	sar	The <code>sar(1M)</code> command provides similar capabilities. <code>swap -s</code> shows the total amount of swap space available on the system.	N
ptx(1)	N		N
pwck(8V)	S		N
pwd(1)	S		N
pwdauthd(8C)	N	Similar capabilities will be available in future releases with unbundled products. See your system vendor for information on this product.	N
quot(8)	S		N
quota(1)	S		N
quotacheck(8)	S		N
quotaoff(8)	S		N
quotaon(8)	S		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
ranlib(1)	C	The ar(1) command automatically provides similar capabilities. ranlib remains as a null script.	N
rarpd(8C)	S		N
rasfilter8to1(1)	N		N
rastrepl(1)	N		N
rc(8)	N	The configuration scripts under /etc/init.d provide similar capabilities. The organization of rc files has changed in SunOS release 5.5 systems. They are now divided by run levels.	N
rc.boot(8)	N	The configuration scripts under /etc/init.d provide similar capabilities.	N
rc.local(8)	N	The configuration scripts under /etc/init.d provide similar capabilities.	N
rcp(1C)	S		N
rdate(8C)	S		N
rdist(1)	S		N
rdump(8)	ufsdump	The ufsdump(1M) command provides similar capabilities.	N
reboot(8)	S		N
red(1)	S		N
refer(1)	S		N
rehash(1)	S		N
remove_brackets(1)	OpenWindows	A version of this command is available with the OpenWindows Text Editor.	N
renice(8)	prionctl	The prionctl(1) command provides similar capabilities.	S
repquota(8)	S		N
reset(1)	A	stty provides similar capabilities.	S
restore(8)	A	The SunOS release 5.5 command, ufsrestore, has been enhanced to take advantage of the end-of-media detection done by ufsdump.	N
rev(1)	N		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
rex(8C)	A	in.rxd provides similar capabilities.	N
rexc(8C)	A	in.rexc provides similar capabilities.	N
rfadmin(8)	N	RFS is not available.	N
rfpasswd(8)	N	RFS is not available.	N
rfstart(8)	N	RFS is not available.	N
rfstop(8)	N	RFS is not available.	N
rfuadmin(8)	N	RFS is not available.	N
rfudaemon(8)	N	RFS is not available.	N
ring_alarm(1)	N		N
rlogin(1C)	C	The ~dsusp sequence for escapes on SunOS release 4.x system software is not available with the SunOS release 5.5 command. Also, the syntax for the -e option differs between the SunOS release 4.x and SunOS release 5.5 commands. In SunOS release 4.x system software, the syntax is -ec; in SunOS release 5.5 system software, it is -e c.	N
rlogind(8C)	in.rlogind		N
rm(1)	S		N
rm_client(8)	A	The admintool(1M) utility replaces this command on SunOS release 5.5 systems.	N
rm_services(8)	A	The swmtool(1M) command provides similar capabilities	N
rmail(8C)	C	The SunOS release 4.x version handles remote mail received using uu(1C). It is explicitly designed for use with uu(1C) and sendmail(8). The SunOS release 5.5 rmail is a link to mail(1) and is a command used for reading mail.	N
rmdel(1)	C	The versions differ in how they treat an unreadable s.file. The SunOS release 4.x command prints an error and continues if it encounters an unreadable s.file. The SunOS release 5.5 command silently ignores the error.	N
rmdir(1)	S		N
rmntstat(8)	N	RFS is not available.	N
rmt(8C)	S		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
roffbib(1)	S		N
route(8C)	C	The SunOS release 4.x route command uses <code>gethostent(3)</code> to look up all symbolic names and gateways, while the SunOS release 5.5 command uses <code>gethostbyname(3)</code> .	N
routed(8)	in.routed		N
rpc.bootparamd(8)	S		N
rpc.etherd(8C)	N	<code>snoop(1m)</code> obsoletes this daemon.	N
rpc.lockd(8C)	lockd		N
rpc.mountd(8C)	mountd		N
rpc.rexd(8C)	S		N
rpc.rquotad(8C)	S		N
rpc.rstatd(8C)	S	Now in <code>/usr/lib/netsvc/rstat</code> .	N
rpc.rusersd(8C)	S	Now in <code>/usr/lib/netsvc/rusers</code> .	N
rpc.rwall(8C)	S	Now in <code>/usr/lib/netsvc/rwall</code> .	N
rpc.showfhd(8C)	showfhd	The <code>showfhd(1M)</code> command provides similar capabilities.	N
rpc.sprayd(8C)	S	Now in <code>/usr/lib/netsvc/spray</code> .	N
rpc.statd(8C)	S	Now in <code>/usr/lib/netsvc/rstat</code> .	N
rpc.user_agend(8C)	N		N
rpc.yppasswdd(8C)	N		N
rpc.yppupdated(8C)	N		N
rpcgen(1)	S		N
rpcinfo(8)	S		N
rrestore(8)	A	The <code>ufsrestore(1M)</code> command provides similar capabilities.	N
rsh(1C)	S		N
runacct(8)	S		N
rup(1C)	S		N
ruptime(1C)	S		N

≡ A

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
rusage(8)	N		S
rusers(1C)	S		N
rwall(1C)	S		N
rwho(1C)	S		N
sa(8)	A	acct (1M) provides similar capabilities.	N
sact(1)	C	The versions differ in how they treat an unreadable s.file. The SunOS release 4.x command will print an error and continue if it encounters an unreadable s.file. The SunOS release 5.5 command silently ignores the error.	N
savecore(8)	S		N
sccs(1)	S		N
sccs-admin(1)	S		N
sccs-cdc(1)	S		N
sccs-comb(1)	S		N
sccs-delta(1)	S		N
sccs-get(1)	S		N
sccs-help(1)	S		N
sccs-prs(1)	S		N
sccs-prt(1)	S		N
sccs-rmdel(1)	S		N
sccs-sact(1)	S		N
sccs-sccsdiff(1)	S		N
sccs-unget(1)	S		N
sccs-val(1)	S		N
sccsdiff(1)	C		N
screenblank(1)	OpenWindows	The OpenWindows xset -s 600 command provides similar capabilities.	N
screendump(1)	N		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
screenload(1)	N		N
script(1)	S		N
scrolldefaults(1)	OpenWindows	The function of this command is now handled by the OpenWindows property window.	N
sdiff(1V) -- SysV	S		N
sed(1V) -- SysV	S		N
sed(1V)	C	The SunOS release 4.x /usr/5bin/sed and the SunOS release 5.5 commands do not strip initial SPACE and TAB characters from text lines.	S
selection_svc(1)	N		N
sendmail(8)	S		N
set4(8)	N		N
set_alarm(1)	N		N
setkeys(1)	N		N
setsid(8V)	N		N
setup_client(8)	N		N
setup_exec(8)	N		N
sh(1)	C	Under SunOS release 4.x system software, the behavior of the builtins echo and test depend on the relative positions of /usr/bin and /usr/5bin in the environment variable PATH. In SunOS release 5.5, the behavior is triggered by the relative ordering of /usr/ueb and /usr/bin.	N
shelltool(1)	OpenWindows	This command is available as an OpenWindows Shell Tool.	N
shift_lines(1)	OpenWindows	An OpenWindows command is available with the OpenWindows Text Editor.	N
showfh(8C)	N		N
showmount(8)	S		N
shutacct(8)	S		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
shutdown(8)	C	The SunOS release 4.x command is very different from the SunOS release 5.5 shutdown(1M) command. By default, the SunOS release 5.5 shutdown(1M) asks for confirmation before starting shutdown activities, while the SunOS release 4.x shutdown(8) does not ask for confirmation. In addition, the following SunOS release 4.x options are not present in the SunOS release 5.5 command: -f, -h, -k, -n, -r.	S
size(1)	C	The SunOS release 4.x command prints sizes in hexadecimal and decimal, and the file name is optional (with a.out as the default). The SunOS release 5.5 command prints them only in decimal, unless the -o or -x option is specified, and the file name is required.	N
skyversion(8)	N		N
sleep(1)	S		N
soelim(1)	S		N
sort(1V) -- SysV	S		N
sort(1V)	C		N
sortbib(1)	S		N
sparc(1)	S		N
spell(1)	C	The SunOS release 4.x -h <i>spellhist</i> option is not available with the SunOS release 5.5 command. This option places misspelled words with a user/date stamp in <i>spellhist</i> .	N
spellin(1)	S		N
spline(1G)	S		N
split(1)	S		N
spray(8C)	C	The SunOS release 4.x-i <i>delay</i> option is not available with the SunOS release 5.5 command. This option specifies that ICMP echo packets should be used rather than RPC.	N
startup(8)	S		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
strings(1)	S		N
strip(1)	S		N
stty(1V) -- SysV	C		N
stty(1V)	C	The following SunOS release 4.x options are not supported by SunOS release 5.5 stty command: decctlq, tandem, cbreak, ctlecho, prterase, crtkill, cols, tab3, crt, dec, term.	S
stty_from_defaults(1)	N		N
su(1V) -- SysV	S		N
su(1V)	C	The SunOS release 4.x -f option is not supported by the SunOS release 4.x /usr/5bin/su or SunOS release 5.5 su command. This option was used for a fast su with csh.	N
sum(1V) -- SysV	S		N
sum(1V)	C		S
sun(1)	S		N
sundiag(8)			N
suninstall(8)	C	The command to install SunOS release 5.5 software is still called suninstall, but the installation procedure has changed completely. See the <i>SPARC: Installing Solaris Software</i> .	N
sunview(1)	OpenWindows	OpenWindows replaces SunView in SunOS release 5.5 systems.	N
sv_acquire(1)	N		N
sv_release(1)	N		N
swapon(8)	swap	The swap(1M) command provides similar capabilities. In general, options to the SunOS release 5.5 swap command replace capabilities of individual swap-related commands, such as swapon, in SunOS release 4.x systems.	N
swin(1)	N		N
switcher(1)	N		N
symorder(1)	S		N
sync(1)	S		N

≡ A

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
sys-unconfig(8)	S		N
syslogd(8)	S		N
t300(1G)	N		S
t300s(1G)	N		S
t4013(1G)	N		S
t450(1G)	N		S
tabs(1V) -- SysV	S		N
tail(1)	S		N
talk(1)	S		N
tar(1)	S		N
tbl(1)	S		N
tcopy(1)	S		N
tcov(1)	N	Available as an unbundled product.	N
tee(1)	S		N
tek(1G)	N		S
tektool(1)	N		N
telnet(1C)	S		N
test(1V) -- SysV	S		N
test(1V)	C		S
textedit(1)	OpenWindows	This command is available as the OpenWindows Text Edit tool.	N
textedit_filters(1)	OpenWindows	An OpenWindows command is available with the OpenWindows Text Editor.	N
tfstd(8)	N		N
tftp(1C)	S		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
tic(8V)	S		N
time(1V) -- SysV	S		N
time(1V)	C	The SunOS release 4.x command provides a different output than the SunOS release 4.x <code>/usr/5bin/time</code> and the SunOS release 5.5 command. The SunOS release 4.x <code>time</code> prints the elapsed time, the time spent in the system, and the time spent executing the command all on one line, instead of on three separate lines.	N
tip(1C)	S		N
toolplaces(1)	N		N
touch(1V) -- SysV	S		N
touch(1V)	C	The SunOS release 4.x <code>-f</code> option is not available. This option attempts to force the <code>touch</code> in spite of read and write permissions on <i>filename</i> .	S
tput(1V) -- SysV	S		N
tr(1V) -- SysV	S		N
tr(1V)	C		S
trace(1)	truss	The <code>truss(1)</code> command provides similar capabilities.	N
traffic(1C)	N		N
troff(1)	S		N
trpt(8C)	N		N
true(1)	S		N
tset(1)	N		S
tsort(1)	S		N
tty(1)	S		N
ttysoftcar(8)	N		N
tunefs(8)	S		N
turnacct(8)	S		N
tvconfig(8)	N		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
tzsetup(8)	N		N
u370(1)	S		N
u3b(1)	S		N
u3b15(1)	S		N
u3b2(1)	S		N
u3b5(1)	S		N
u1(1)	S		N
umask(1)	S		N
umount(8)	C	The interface differs significantly between the two versions. In the SunOS release 5.5 command, most options are changed and must be supplied as file system-specific options.	N
umount_tfs(8)	N		N
unadv(8)	N	RFS not available.	N
uname(1)	S		N
uncompress(1)	S		N
unconfigure(8)	N		N
unexpand(1)	S		N
unget(1)	C	The versions differ in how they treat an unreadable s.file. The SunOS release 4.x version will print an error and continue if it encounters an unreadable s.file. The SunOS release 5.5 version silently ignores the error.	N
unifdef(1)	S		N
uniq(1)	S		N
units(1)	S		N
unix2dos(1)	S		N
unlink(8V)	S		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
unpack(1V) -- SysV	S		N
unpack(1V) -- SysV	C	With the SunOS release 4.x /usr/5bin/pack command, file names are restricted to 12 characters. In SunOS release 5.5 system software, they are restricted to {NAME_MAX} - 2. The SunOS release 5.5 pack and unpack commands are compatible with the SunOS release 4.x commands.	
unwhiteout(1)	N		N
update(8)	fsflush	The fsflush process provides this capability.	N
uptime(1)	who	The w - u command provides similar capabilities.	S
users(1)	who	The who -q provides similar capabilities.	S
ustar(1V)	tar	The tar(1) command provides similar capabilities.	N
uuccheck(8C)	S		N
uucico(8C)	S		N
uucleanup(8C)	S		N
uucp(1C)	S		N
uudecode(1C)	S		N
uuencode(1C)	S		N
uulog(1C)	C	The -u option, which allows printing of information about work done for a specified username, is no longer supported.	N
uuname(1C)	S		N
uupick(1C)	S		N
uusched(8C)	S		N
uusend(1C)	N		N
uustat(1C)	S		N
uuto(1C)	S		N
uux(1C)	S		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
uuxqt(8C)	S		N
vacation(1)	S		N
val(1)	S		N
vax(1)	S		N
vedit(1)	S		N
vfontinfo(1)	N		N
vgrind(1)	S		N
vi(1)	S		N
view(1)	S		N
vipw(8)	N		S
vmstat(8)	C	The -f option is no longer available.	N
vplot(1)	N		S
vswap(1)	N		N
vtroff(1)	N		N
vwidth(1)	N		N
w(1)	S		N
wait(1)	S		N
wall(1)	S		N
wc(1)	S		N
what(1)	S		N
whatis(1)	C		N
whereis(1)	N		S
which(1)	S		N
who(1)	S		N
whoami(1)	id	The id(1) command provides similar capabilities. The id command prints the user name and user and group IDs, instead of just the user name.	S

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
whois(1)	S		N
write(1)	S		N
xargs(1V) -- SysV	S		N
xget(1)	N		N
xsend(1a)	N		N
xstr(1)	S		N
yacc(1)	S		N
yes(1)	N		N
ypbatchupd(8C)	N		N
ypbind(8)	S	Now in /usr/lib/netsvc/yp.	N
ypcat(1)	S		N
ypinit(8)	S		N
ypmatch(1)	S		N
yppasswd(1)	S	The yppasswd command is still available on SunOS release 5.5 systems to access the password information on NIS servers. The equivalent command for NIS+ databases is nispasswd(1). The passwd(1) command can handle passwords in all supported databases (NIS, NIS+, files).	N
yppoll(8)	S		N
yppush(8)	N		N
ypserv(8)	N		N
ypset(8)	S		N
ypupdated(8C)	N		N
ypwhich(8)	S		N
ypxfr(8)	S	Now in /usr/lib/netsvc/yp.	N
ypxfrd(8)	S		N



Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.5	Alternative Available and Notes	BSD
zcat(1)	S		N
zdump(8)	S		N
zic(8)	S		N

System Calls Reference Table



This appendix contains the System Calls reference table. This table lists all SunOS release 4.x system calls, and shows their status in the following environments: Solaris 2.5, the ABI, the SVID, SVR4, and the SunOS/BSD Source Compatibility Package.

Using the Reference Table

- If an interface is listed as “changed” (C), a brief description of differences between the SunOS release 4.x system call and the Solaris 2.5 system call is provided.
- If an interface is listed as “the same” (S), the Solaris 2.5 interface will support all features of the SunOS release 4.x interface. In some cases the interface has been enhanced, but can be considered a complete superset of the SunOS release 4.x interface. Note, many system calls are now available as library routines. The Notes column will show the new routine man page reference.
- If an interface has an “alternative” (A), check the Notes section for its replacement.
- If an interface is listed as “not available” (N), you cannot use that interface.

- If the interface includes `errno` values that are not supported in the standard, it is indicated with "#". `errno` differences do not necessarily break compatibility. Note that although `EDQUOT`, `EFAULT`, and `EIO` are often not listed with ABI or SVID, these `errno` values are supported by an ABI or SVID compliant system if appropriate.

SunOS release 4.x offers a System V Software installation option that provides System V compatible versions of many utilities, system calls, and library routines. The System V interfaces are included in the following tables. When referring to the System V version of a SunOS release 4.x interface, the string 'SysV' is appended to the interface.

For complete information on all Solaris 2.5 interfaces, see the *man Pages(2): System Calls*.

Examples

Below are sample table entries followed by an interpretation of the table entry.

SunOS release 4.x System Call	SunOS 5.5	Alternative Available and Notes	ABI	SVID	SVR4	BSD
<code>mctl(2)</code>	A	The <code>memcntl(2)</code> system call provides similar functionality.	A	A	A	S

The `mctl()` system call is not available in the ABI, SVID, SVR4, or the SunOS release 5.5. Any applications that use this system call should be rewritten to use the `memcntl()` call. A version of `mctl()` is available with the SunOS/BSD Compatibility package, but applications that use it will not be compatible with other SVR4 systems.

SunOS release 4.x System Call	SunOS 5.5	Alternative Available and Notes	ABI	SVID	SVR4	BSD
<code>getsockname(2)</code>	S#	The <code>errno</code> value <code>ENOBUFS</code> used by SunOS release 4.x <code>getsockname()</code> system call has been changed to <code>ENOSR</code> in the SVR4 and SunOS 5.5 version.	N	N	S#	N

The `getsockname()` system call is not defined in the ABI, or SVID. The `getsockname()` call in SunOS release 5.5 and SVR4 releases is the same as the one in SunOS release 4.x, except SunOS release 5.5 sets `errno` to `ENOSR` for the error condition that caused SunOS release 4.x to set `errno` to `ENOBUFS`.

System Calls

Table B-1 System Calls Reference Table

SunOS Release 4.x System Call	SunOS Re-lease 5.5	Notes	ABI	SVID	SVR4	BSD
WEXITSTATUS(2)	C	The <i>union wait</i> , supported in SunOS release 4.x for backwards compatibility with previous SunOS releases, is not supported in the SVR4 and SunOS release 5.5 versions.	N	N	S	S
WIFEXITED(2)	C	The <i>union wait</i> , supported in SunOS release 4.x for backwards compatibility with previous SunOS releases, is not supported in the SVR4 and SunOS release 5.5 versions.	N	N	C	S
WIFSIGNALED(2)	C	The <i>union wait</i> , supported in SunOS release 4.x for backwards compatibility with previous SunOS releases, is not supported in the SVR4 and SunOS release 5.5 versions.	N	N	C	S
WIFSTOPPED(2)	C	The <i>union wait</i> , supported in SunOS release 4.x for backwards compatibility with previous SunOS releases, is not supported in the SVR4 and SunOS release 5.5 versions.	N	N	C	S
WSTOPSIG(2)	C	The <i>union wait</i> , supported in SunOS release 4.x for backwards compatibility with previous SunOS releases, is not supported in the SVR4 and SunOS release 5.5 versions.	N	N	C	S
WTERMSIG(2)	C	The <i>union wait</i> , supported in SunOS release 4.x for backwards compatibility with previous SunOS releases, is not supported in the SVR4 and SunOS release 5.5 versions.	N	N	C	S
_exit(2V) — SysV	S		S	S	S	N
accept(2)	S	Now accept(3N).	N	N	S	N
access(2V) — SysV	S		S	S	S	N
acct(2)	C#	The following symbolic names are valid for the <code>acct</code> structure member <code>ac_flag</code> (defined in <code><sys/acct.h></code>) for SunOS release 4.x version, but not for SunOS release 5.5, ABI, SVID, and SVR4 versions: <code>ACOMPAT</code> , <code>ACORE</code> , <code>AXSIG</code> . Also, the accounting record format differs between SunOS release 4.x and SunOS release 5.5, ABI, SVID, and SVR4 versions.	C#	C#	C#	N
adjtime(2)	S		N	S	S	N
async_daemon(2)	N		N	N	N	N

≡ B

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Re-lease 5.5	Notes	ABI	SVID	SVR4	BSD
audit(2)	N		N	N	N	N
auditon(2)	N		N	N	N	N
auditsvc(2)	N		N	N	N	N
bind(2)	S	Now bind(3N).	N	N	S	N
brk(2)	S		N	N	S	N
chdir(2V) — SysV	S		S	S	S	N
chmod(2V) — SysV	C#	The following symbolic access modes (<sys/stat.h>) are supported by SunOS release 4.x chmod function but not by SunOS release 5.5, ABI, SVID, or SVR4 versions: S_IREAD (00400), S_IWRITE (00200), S_IEXEC (00100). However, the equivalent SunOS release 5.5, ABI, SVID, or SVR4 symbolic access modes, S_IRUSR (00400), S_IWUSR (00200), S_IXUSR (00100), have the same meanings.	C#	C#	C#	N
chown(2V)	C	In the SunOS release 4.x version, the <i>owner</i> and <i>group</i> arguments of <code>chown()</code> are of type <code>int</code> . In SunOS release 5.5, ABI, SVID, and SVR4, <i>owner</i> is of type <code>uid_t</code> , and <i>group</i> is of type <code>gid_t</code> . In the SunOS release 4.x version, if the final component of <i>path</i> is a symbolic link, the ownership of the symbolic link was changed. In the SunOS 5.5 version, <code>chown()</code> changes the ownership of the file or directory referred to by the symbolic link. Use <code>lchown(2)</code> in SunOS release 5.5 to change the ownership of a symbolic link.	C	C	C	N
chown(2V) — SysV	S		S	S	S	N
chroot(2)	S		S	S	S	N
close(2V) — SysV	S		S	S	S	N
connect(2)	S#	Now connect(3N).	N	N	S#	N

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Re-lease 5.5	Notes	ABI	SVID	SVR4	BSD
<code>creat(2V)</code>	C#	In SunOS release 4.x, the <i>mode</i> argument to <code>creat()</code> is of type <code>int</code> , while in SunOS release 5.5, ABI, SVID, and SVR4, the <i>mode</i> argument is of type <code>mode_t</code> . Also, SunOS release 5.5, ABI, SVID, and SVR4 versions include <code><fcntl.h></code> while SunOS release 4.x version does not. The following symbolic access modes (<code><sys/stat.h></code>) are supported by SunOS release 4.x version of <code>creat()</code> , but not by SunOS release 5.5, ABI, SVID, or SVR4 versions: <code>S_IREAD</code> (00400), <code>S_IWRITE</code> (00200), <code>S_IEXEC</code> (00100). However, the equivalent SunOS release 5.5, ABI, SVID, and SVR4 symbolic access modes, <code>S_IRUSR</code> (00400), <code>S_IWUSR</code> (00200), <code>S_IXUSR</code> (00100), do have the same definitions, are defined in SunOS release 4.x <code><sys/stat.h></code> , and thus should be used. The following <code>errno</code> flags are valid for SunOS release 4.x version of this system call but are not valid in SunOS release 5.5, ABI, SVID, or SVR4 versions: <code>ENXIO</code> , <code>EOPNOTSUPP</code> .	C#	C#	C#	N
<code>creat(2V) — SysV</code>	C#	The following symbolic access modes (<code><sys/stat.h></code>) are supported by the SunOS release 4.x version of <code>creat()</code> , but not by SunOS release 5.5, ABI, SVID, or SVR4 versions: <code>S_IREAD</code> (00400), <code>S_IWRITE</code> (00200), <code>S_IEXEC</code> (00100). However, the equivalent SunOS release 5.5, ABI, SVID, and SVR4 symbolic access modes, <code>S_IRUSR</code> (00400), <code>S_IWUSR</code> (00200), <code>S_IXUSR</code> (00100), do have the same definitions, are defined in SunOS release 4.x <code><sys/stat.h></code> , and thus should be used. The following <code>errno</code> flags are valid for the SunOS release 4.x version of this system call but are not valid in SunOS release 5.5, ABI, SVID, or SVR4 versions: <code>ENXIO</code> , <code>EOPNOTSUPP</code> .	C#	C#	C#	N
<code>dup(2V) — SysV</code>	S		S	S	S	N
<code>dup2(2V) — SysV</code>	S	Now <code>dup2(3C)</code> .	S	S	S	N
<code>execve(2V) — SysV</code>	S		S	S	S	N
<code>fchdir(2V) — SysV</code>	S		S	S	S	N

≡ B

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Re-lease 5.5	Notes	ABI	SVID	SVR4	BSD
<code>fchmod(2V)</code> — SysV	C	The following symbolic access modes (<code><sys/stat.h></code>) are supported by the SunOS release 4.x version of <code>fchmod()</code> , but not by SunOS release 5.5, ABI, SVID, or SVR4 versions: <code>S_IREAD</code> (00400), <code>S_IWRITE</code> (00200), <code>S_IEXEC</code> (00100). However, the equivalent SunOS release 5.5, ABI, SVID, and SVR4 symbolic access modes, <code>S_IRUSR</code> (00400), <code>S_IWUSR</code> (00200), <code>S_IXUSR</code> (00100), do have the same definitions, are defined in SunOS release 4.x <code><sys/stat.h></code> , and thus should be used.	C	C	C	N
<code>fchown(2)</code>	S		S	S	S	N
<code>fchroot(2)</code>	S		N	N	N	N
<code>fcntl(2V)</code> — SysV	C	In SunOS release 4.x, the following flags are valid for the <code>F_SETFL</code> command: <code>O_APPEND</code> , <code>O_SYNC</code> , and <code>O_NDELAY</code> , and the <code>FSYNC</code> , <code>FNDELAY</code> , and <code>FNBBIO</code> flags defined in <code><sys/file.h></code> . SunOS release 5.5, ABI, SVID, and SVR4 support only the <code>O_APPEND</code> , <code>O_SYNC</code> , <code>O_NDELAY</code> , and <code>O_NONBLOCK</code> flags, thus, <code>O_SYNC</code> should be used in place of <code>FSYNC</code> , and <code>O_NONBLOCK</code> should be used in place of <code>FNDELAY</code> and <code>FNBBIO</code> . <code>O_NONBLOCK</code> should also be used in place of <code>O_NDELAY</code> , which is being phased out. SunOS release 4.x <code>F_GETOWN</code> and <code>F_SETOWN</code> commands are not supported in SunOS release 5.5, ABI, SVID, or SVR4 versions.	C	C	C	N
<code>flock(2)</code>	N		N	N	N	S
<code>fork(2V)</code>	C	In SunOS release 4.x, <code>fork()</code> returns a value of type <code>int</code> . In SunOS release 5.5, ABI, SVID, and SVR4, <code>fork()</code> returns a value of type <code>pid_t</code> . Also, SunOS release 5.5, ABI, SVID, or SVR4 versions includes <code><unistd.h></code> and <code><sys/types.h></code> while the SunOS release 4.x version does not.	C	C	C	N
<code>fork(2V)</code> — SysV	S		S	S	S	N
<code>fpathconf(2V)</code> — SysV	S		S	S	S	N
<code>fstat(2V)</code> — SysV	S		S	S	S	N

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Re-lease 5.5	Notes	ABI	SVID	SVR4	BSD
<code>fstatfs(2)</code>	A	The <code>fstatvfs(2)</code> system call provides equivalent functionality.	A	A	A	S
<code>fsync(2)</code>	S		S	S	S	N
<code>ftruncate(2)</code>	S	Now <code>ftruncate(3C)</code> .	N	N	S	N
<code>getauid(2)</code>	N		N	N	N	N
<code>getdents(2)</code>	S		N	N	S	N
<code>getdirenties(2)</code>	A	The <code>getdents(2)</code> system call provides equivalent functionality.	N	N	N	N
<code>getdomainname(2)</code>	A	The <code>sysinfo(2)</code> system call provides equivalent functionality.	N	N	N	N
<code>getdtablesize(2)</code>	A	The <code>getrlimit(2)</code> system call with the <i>resource</i> argument set to <code>RLIMIT_NOFILE</code> provides similar functionality.	A	A	A	S
<code>getegid(2V)</code>	C	In SunOS release 4.x, <code>getegid()</code> returns a value of type <code>int</code> . In SunOS release 5.5, ABI, SVID, and SVR4, <code>getegid()</code> returns a value of type <code>gid_t</code> . Also, SunOS release 5.5, ABI, SVID, or SVR4 versions includes <code><unistd.h></code> and <code><sys/types.h></code> while the SunOS release 4.x version does not.	C	C	C	N
<code>getegid(2V) — SysV</code>	S		S	S	S	N
<code>geteuid(2V)</code>	C	In SunOS release 4.x, <code>geteuid()</code> returns a value of type <code>int</code> . In SunOS release 5.5, ABI, SVID, and SVR4, <code>geteuid()</code> returns a value of type <code>uid_t</code> . Also, SunOS release 5.5, ABI, SVID, and SVR4 versions include <code><unistd.h></code> and <code><sys/types.h></code> while the SunOS release 4.x version does not.	C	C	C	N
<code>geteuid(2V) — SysV</code>	S		S	S	S	N

≡ B

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Re-lease 5.5	Notes	ABI	SVID	SVR4	BSD
getgid(2V)	C	In SunOS release 4.x, <code>getgid()</code> returns a value of type <code>int</code> . In SunOS release 5.5, ABI, SVID, and SVR4, <code>getgid()</code> returns a value of type <code>gid_t</code> . Also, SunOS release 5.5, ABI, SVID, and SVR4 versions include <code><unistd.h></code> and <code><sys/types.h></code> while the SunOS release 4.x version does not.	C	C	C	N
getgid(2V) — SysV	S		S	S	S	N
getgroups(2V)	C	In SunOS release 4.x, the <code>gidset</code> argument to <code>getgroups()</code> is of type <code>int</code> , while in SunOS release 5.5, ABI, SVID, and SVR4, the <code>grouplist</code> argument is of type <code>gid_t</code> . Also, SunOS release 5.5, ABI, SVID, and SVR4 versions include <code><unistd.h></code> and <code><sys/types.h></code> while the SunOS release 4.x version does not.	C	C	C	N
getgroups(2V) — SysV	S		S	S	S	N
gethostid(2)	A	Now <code>gethostid(3)</code> . The <code>sysinfo(2)</code> system call with the <code>command</code> argument set to <code>SI_HW_SERIAL</code> provides similar functionality.	N	N	N	S
gethostname(2)	A	Now <code>gethostname(3)</code> . The <code>sysinfo(SI_HOSTNAME, name, namelen)</code> ; routine provides similar functionality.	N	N	N	S
getitimer(2)	S		N	S	S	N
getmsg(2)	S		S	S	S	N
getpagesize(2)	A	Now <code>getpagesize(3)</code> . The <code>sysconf(3C)</code> routine provides similar functionality.	A	A	A	S
getpeername(2)	S#	Now <code>getpeername(3N)</code> . The following <code>errno</code> flag is valid for SunOS release 4.x <code>getpeername()</code> system call but is not valid in the SVR4 and SunOS 5.5 version: <code>ENOBUFS</code> .	N	N	S#	N
getpgid(2V)	S		S	S	S	N

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Release 5.5	Notes	ABI	SVID	SVR4	BSD
getpgrp(2V)	C	The SunOS release 4.x version of <code>getpgrp()</code> has an argument <code>pid</code> , and <code>getpgrp()</code> returns the process group of the process indicated by <code>pid</code> . SunOS release 5.5, ABI, SVID, and SVR4 versions of <code>getpgrp()</code> do not accept an argument, and <code>getpgrp()</code> returns the process group ID of the calling process. Also, SunOS release 4.x <code>getpgrp()</code> returns a value of type <code>int</code> , while SunOS release 5.5, ABI, SVID, and SVR4 <code>getpgrp()</code> return a value of type <code>pid_t</code> . SunOS release 5.5, ABI, SVID, and SVR4 versions include <code><unistd.h></code> and <code><sys/types.h></code> while the SunOS release 4.x version does not.	C	C	C	N
getpgrp(2V) — SysV	S		S	S	S	N
getpid(2V)	C	In SunOS release 4.x, <code>getpid()</code> returns a value of type <code>int</code> . In SunOS release 5.5, ABI, SVID, and SVR4, <code>getpid()</code> returns a value of type <code>pid_t</code> . Also, SunOS release 5.5, ABI, SVID, and SVR4 versions include <code><unistd.h></code> and <code><sys/types.h></code> while the SunOS release 4.x version does not.	C	C	C	N
getpid(2V) — SysV	S		S	S	S	N
getppid(2V)	C	In SunOS release 4.x, <code>getppid()</code> returns a value of type <code>int</code> . In SunOS release 5.5, ABI, SVID, and SVR4, <code>getppid()</code> returns a value of type <code>pid_t</code> . Also, SunOS release 5.5, ABI, SVID, and SVR4 versions include <code><unistd.h></code> and <code><sys/types.h></code> while the SunOS release 4.x version does not.	C	C	C	N
getppid(2V) — SysV	S		S	S	S	N
getpriority(2)	A	Now <code>getpriority(3)</code> . The <code>pricntl(2)</code> system call provides similar functionality.	A	A	A	S

≡ B

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Re-lease 5.5	Notes	ABI	SVID	SVR4	BSD
getrlimit(2)	C	In SunOS release 4.x, RLIMIT_RSS is a supported resource (the maximum size, in bytes, to which a process's resident set size may grow) which is not supported in SunOS release 5.5, ABI, SVID, and SVR4 versions. SunOS release 5.5, ABI, SVID, and SVR4 versions additionally support the RLIMIT_AS resource, the maximum amount of a process's address space that is defined (in bytes). Also, SunOS release 5.5, ABI, SVID, and SVR4 versions of <code>rlim_cur</code> (current soft limit) and <code>rlim_max</code> (hard limit) fields in the <code>rlimit</code> structure are <code>rlim_t</code> rather than <code>int</code> as in SunOS release 4.x.	C	C	C	N
getrusage(2)	A	Now <code>getusage(3C)</code> .	N	N	N	C
getsockname(2)	S#	The <code>errno</code> value <code>ENOBUFFS</code> used by SunOS release 4.x <code>getsockname()</code> system call has been changed to <code>ENOSR</code> in the SVR4 and SunOS 5.5 version.	N	N	S#	N
getsockopt(2)	S	Now <code>getsockopt(3N)</code> .	N	N	S	N
gettimeofday(2)	S	Now <code>gettimeofday(3C)</code> .	N	S	S	S
getuid(2V)	C	In SunOS release 4.x, <code>getuid()</code> returns a value of type <code>int</code> . In SunOS release 5.5, ABI, SVID, and SVR4, <code>getuid()</code> returns a value of type <code>uid_t</code> . Also, SunOS release 5.5, ABI, SVID, and SVR4 versions include <code><unistd.h></code> and <code><sys/types.h></code> while the SunOS release 4.x version does not.	C	C	C	N
getuid(2V) — SysV	S		S	S	S	N
ioctl(2)	C	See “ <code>ioctl()</code> Requests” on page 157	C	C	C	N

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Re-lease 5.5	Notes	ABI	SVID	SVR4	BSD
kill(2V)	C	In SunOS release 4.x, if a signal is sent to a group of processes (as with, if <i>pid</i> is 0 or negative), and if the process sending the signal is a member of that group, the signal is not sent to the sending process as well. In SunOS release 5.5, ABI, SVID, and SVR4, the signal is sent to the sending process as well. In SunOS release 4.x, the <i>pid</i> argument is of type <code>int</code> , while in SunOS release 5.5, ABI, SVID, and SVR4, the <i>pid</i> argument is of type <code>pid_t</code> . Also, SunOS release 5.5, ABI, SVID, and SVR4 versions include <code><sys/types.h></code> while the SunOS release 4.x version does not.	C	C	C	N
kill(2V) — SysV	S		S	S	S	N
killpg(2)	A	The <code>kill(2)</code> system call provides similar functionality. Replace <code>killpg(pgrp, sig)</code> with <code>kill(-pgrp, sig)</code> .	A	A	A	S
link(2V) — SysV	C	In SunOS release 5.5, ABI, SVID, and SVR4 version of <code>link()</code> , if the last component of the first argument is a symbolic link, it will not be followed and a hard link will be made to the symbolic link.	C	C	C	N
listen(2)	S	Now <code>listen(3N)</code> .	N	N	S	N
lseek(2V) — SysV	S		S	S	S	N
lstat(2V) — SysV	S		S	S	S	N
mctl(2)	A	The <code>memcntl(2)</code> system call provides similar functionality.	A	A	A	S
mincore(2)	C	In SunOS release 4.x, argument <i>len</i> is of type <code>int</code> , while in SVR4 and SunOS 5.5, argument <i>len</i> is of type <code>size_t</code> which is defined to be <code>unsigned int</code> . SunOS release 5.5 version also requires inclusion of <code><unistd.h></code> .	N	N	C	N

≡ B

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Release 5.5	Notes	ABI	SVID	SVR4	BSD
<code>mkdir(2V)</code>	C	In SunOS release 4.x, the mode argument is of type <code>int</code> , while in SunOS release 5.5, ABI, SVID, and SVR4, the mode argument is of type <code>mode_t</code> . Also, SunOS release 5.5, ABI, SVID, and SVR4 versions include <code><sys/types.h></code> and <code><sys/stat.h></code> while the SunOS release 4.x version does not. The following symbolic access modes (<code><sys/stat.h></code>) are supported by the SunOS release 4.x version of <code>mkdir()</code> , but not by SunOS release 5.5, ABI, SVID, and SVR4 versions: <code>S_IREAD</code> (00400), <code>S_IWRITE</code> (00200), <code>S_IEXEC</code> (00100). However, the equivalent SunOS release 5.5, ABI, SVID, and SVR4 symbolic access modes, <code>S_IRUSR</code> (00400), <code>S_IWUSR</code> (00200), <code>S_IXUSR</code> (00100), do have the same definitions, are defined in SunOS release 4.x <code><sys/stat.h></code> , and thus should be used.	C	C	C	N
<code>mkdir(2V)</code> — SysV	C	The following symbolic access modes (<code><sys/stat.h></code>) are supported by the SunOS release 4.x version of <code>mkdir()</code> , but not by SunOS release 5.5, ABI, SVID, and SVR4 versions: <code>S_IREAD</code> (00400), <code>S_IWRITE</code> (00200), <code>S_IEXEC</code> (00100). However, the equivalent SunOS release 5.5, ABI, SVID, and SVR4 symbolic access modes, <code>S_IRUSR</code> (00400), <code>S_IWUSR</code> (00200), <code>S_IXUSR</code> (00100), do have the same definitions, are defined in SunOS release 4.x <code><sys/stat.h></code> , and thus should be used.	C	C	C	N
<code>mkfifo(2V)</code> — SysV	S	Now <code>mkfifo(3C)</code> .	S	S	S	N

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Release 5.5	Notes	ABI	SVID	SVR4	BSD
mknod(2V) — SysV	C	The mode argument to <code>mknod()</code> is of type <code>int</code> in SunOS release 4.x and of type <code>mode_t</code> in SunOS release 5.5, ABI, SVID, and SVR4. The <code>dev</code> argument is of type <code>int</code> in SunOS release 4.x and of type <code>dev_t</code> in SunOS release 5.5, or the ABI, SVID, or SVR4. The following symbolic access modes (<code><sys/stat.h></code>) are supported by the SunOS release 4.x version of <code>mknod()</code> , but not by SunOS release 5.5, ABI, SVID, and SVR4 versions: <code>S_IREAD</code> (00400), <code>S_IWRITE</code> (00200), <code>S_IEXEC</code> (00100). However, the equivalent symbolic access modes, <code>S_IRUSR</code> (00400), <code>S_IWUSR</code> (00200), <code>S_IXUSR</code> (00100), do have the same definitions, are defined in SunOS release 4.x <code><sys/stat.h></code> , and thus should be used.	C	C	C	N
<code>mmap(2)</code>	C	In SunOS release 4.x, <code>mmap</code> <i>flag</i> option value includes <code>MAP_TYPE</code> , defined in <code><sys/mman.h></code> , which is not defined in SunOS release 5.5, ABI, SVID, and SVR4 <code><sys/mman.h></code> .	C	C	C	N

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Re-lease 5.5	Notes	ABI	SVID	SVR4	BSD
mount(2)	C#	<p>The SunOS release 4.x version of mount() and the SunOS 5.5 release, or the ABI, SVID, or SVR4 version of mount() are incompatible in a number of respects. The first argument in SunOS release 4.x, <i>type</i>, is the file system type name, while in SunOS release 5.5, ABI, SVID, and SVR4, the first argument is <i>fs</i>, is the name of the file system. In SunOS release 5.5, ABI, SVID, and SVR4 versions, the file system type name, <i>fstype</i>, is the fourth argument to mount(). The SunOS release 4.x version uses a single parameter (<i>caddr_t data</i>, the fourth argument) to pass type specific arguments, while SunOS release 5.5, ABI, SVID, and SVR4 version uses two parameters (five and six: <code>const char *dataptr</code> and <code>int datalen</code>). Also, SunOS release 5.5, ABI, SVID, and SVR4 versions include <code><sys/types.h></code> before <code><sys/mount.h></code> while the SunOS release 4.x version does not.</p> <p>The SunOS release 4.x version of <code><sys/mount.h></code> defines symbolic constants for the mount() <i>flags</i> argument (M_NEWTYPE, M_RDONLY, M_NOSUID, M_NEWTYPE, M_GRPID, M_REMOUNT, M_NOSUB, M_MULTTI) which are not defined in SunOS release 5.5, or the ABI, SVID, or SVR4 <code><sys/mount.h></code>. Instead, replace M_RDONLY with MS_RDONLY, M_NOSUID with MS_NOSUID, and M_REMOUNT with MS_REMOUNT. The M_NEWTYPE flag is specific to the SunOS release 4.x version of mount() and no replacement is required for SunOS release 5.5, ABI, SVID, or SVR4. The functionality of the following flags, defined in <code><sys/mount.h></code>, is not supported by the SunOS 5.5 release, or the ABI, SVID, or SVR4 versions: M_NOSUB, M_GRPID, M_MULTTI.</p> <p>SunOS release 4.x mount() uses the following <code>errno</code> values which are not returned by the SunOS 5.5 release, or the ABI, SVID, or SVR4 version: ENODEV, EACCES, EMFILE, ENOMEM.</p>	C#	C#	C#	N
mprotect(2)	S		S	S	S	N
msgctl(2)	S		S	S	S	N
msgget(2)	S		S	S	S	N

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Re-lease 5.5	Notes	ABI	SVID	SVR4	BSD
<code>msgrcv(2)</code>	S		S	S	S	N
<code>msgsnd(2)</code>	S		S	S	S	N
<code>msync(2)</code>	S		S#	S#	S	N
<code>munmap(2)</code>	S		S	S	S	N
<code>nfssvc(2)</code>	A	This interface is replaced in SunOS release 5.5 by the <code>nfssys(NFS_SVC, ...)</code> routine.	N	N	N	N
<code>open(2V)</code>	C#	<p>The <i>mode</i> argument to <code>open()</code> is of type <code>int</code> in SunOS release 4.x and of type <code>mode_t</code> in SunOS release 5.5, ABI, SVID, and SVR4. In SunOS release 4.x, if the <i>path</i> argument is an empty string, the kernel maps this empty pathname to <code>`.`</code>, the current directory. In SunOS release 5.5, ABI, SVID, and SVR4, if <i>path</i> points to an empty string an error results. In SunOS release 4.x, if the <code>O_NDELAY</code> or <code>O_NONBLOCK</code> flag is set on a call to <code>open</code>, only the <code>open()</code> call itself is effected. In SunOS release 5.5, ABI, SVID, and SVR4, if the <code>O_NDELAY</code> or <code>O_NONBLOCK</code> flag is set on a call to <code>open()</code>, the corresponding flag is set for that file descriptor and subsequent reads and writes to that descriptor will not block.</p> <p>Also, SunOS release 5.5, ABI, SVID, and SVR4 versions include <code><sys/types.h></code> and <code><sys/stat.h></code> while the SunOS release 4.x version does not.</p> <p>The following <code>errno</code> value is valid for the SunOS 4.1 version of this system call but is not returned in SunOS release 5.5, ABI, SVID, and SVR4 versions: <code>EOPNOTSUPP</code>.</p>	C#	C#	C#	N
<code>open(2V) — SysV</code>	S#	The following <code>errno</code> value is valid for the SunOS release 4.x version of this system call but is not returned in SunOS release 5.5, ABI, SVID, and SVR4 versions: <code>EOPNOTSUPP</code> .	S#	S#	S#	N
<code>pathconf(2V) — SysV</code>	S		S	S	S	N
<code>pipe(2V) — SysV</code>	S		S	S	S	N
<code>poll(2)</code>	S		S	S	S	N
<code>profil(2)</code>	S		S	S	S	N

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Re-lease 5.5	Notes	ABI	SVID	SVR4	BSD
<code>ptrace(2)</code>	C#	The optional <code>addr2</code> argument to SunOS release 4.x <code>ptrace()</code> system call is not supported by the SunOS 5.5 routine. The <code>request</code> argument to <code>ptrace()</code> is of type <code>enum ptracereq</code> in SunOS release 4.x and of type <code>int</code> in SunOS release 5.5. The <code>pid</code> argument to <code>ptrace()</code> is of type <code>int</code> in SunOS release 4.x and of type <code>pid_t</code> in SunOS release 5.5. Also, the SunOS 5.5 version includes <code><sys/types.h></code> while the SunOS release 4.x version includes <code><signal.h></code> , <code><sys/ptrace.h></code> , and <code><sys/wait.h></code> . The following <code>errno</code> flag is valid for the SunOS release 4.x version of this system call, but is not valid in the SunOS 5.5 version: <code>EPERM</code> . See “ <code>ptrace()</code> Request Values” on page 159 for information on valid <i>request</i> values.	C#	C#	C#	N
<code>putmsg(2)</code>	S		S	S	S	N
<code>quotactl(2)</code>	A	The <code>Q_QUOTACTL</code> <code>ioctl</code> system call provides similar functionality.	A	A	A	N
<code>read(2V)</code>	C#	The following <code>errno</code> flags are valid for the SunOS release 4.x version of this system call but are not valid in SunOS release 5.5, ABI, SVID, and SVR4 versions: <code>EISDIR</code> , <code>EWOULDBLOCK</code> .	C#	C#	C#	N
<code>read(2V) — SysV</code>	C#	The <code>nbyte</code> argument to <code>read()</code> is of type <code>int</code> in SunOS release 4.x and of type <code>unsigned</code> in SunOS release 5.5. The SunOS 5.5 <code>read()</code> system call does not support BSD 4.2 style non-blocking I/O (with the <code>FIONBIO</code> <code>ioctl()</code> request or a call to <code>fcntl(2V)</code> using the <code>FNDELAY</code> flag from <code><sys/file.h></code> or the <code>O_NDELAY</code> flag from <code><fcntl.h></code> in the 4.2BSD environment) as does SunOS release 4.x <code>read()</code> routine. The following <code>errno</code> flags are valid for the SunOS release 4.x version of this system call, but are not valid in the SunOS 5.5 version: <code>EISDIR</code> , <code>EWOULDBLOCK</code> .	C#	C#	C#	N
<code>readlink(2)</code>	S		S	S	S	N

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Re-lease 5.5	Notes	ABI	SVID	SVR4	BSD
readv(2V)	C#	The following <code>errno</code> flags are valid for the SunOS release 4.x version of this system call but are not valid in SunOS release 5.5, ABI, SVID, and SVR4 versions: <code>EISDIR</code> , <code>EWOULDBLOCK</code> .	C#	C#	C#	N
readv(2V) — SysV	C#	SunOS release 4.x and SunOS release 5.5, or the SVID or SVR4 <code>iovec</code> structures (defined in <code><sys/uio.h></code>) differ slightly. SunOS release 4.x <code>iovec</code> 's <code>iov_len</code> field is defined as integer, while SunOS release 5.5, or the SVID or SVR4 <code>iov_len</code> is defined as unsigned. SunOS release 5.5, or the SVID or SVR4 <code>readv()</code> system call does not support BSD 4.2 style non-blocking I/O as does SunOS release 4.x	C#	C#	C#	N
reboot(2)	A	Now <code>reboot(3C)</code> . The <code>uadmin(2)</code> system call provides similar functionality.	N	N	N	S
recv(2)	S	Now <code>recv(3N)</code> .	N	N	S	N
recvfrom(2)	S	Now <code>recvfrom(3N)</code> .	N	N	S	N
recvmsg(2)	S	Now <code>recvmsg(3N)</code> .	N	N	S	N
rename(2V) — SysV	S#	The SunOS release 5.5, ABI, SVID, and SVR4 versions include <code><unistd.h></code> while the SunOS release 4.x version does not. The following <code>errno</code> flag is valid for the SunOS release 4.x version of this system call but is not valid in SunOS release 5.5, ABI, SVID, and SVR4 versions: <code>ENOTEMPTY</code> . SunOS release 5.5, ABI, SVID, and SVR4 versions set <code>errno</code> to flag <code>EEXIST</code> instead.	S#	S#	S#	N
rmdir(2V) — SysV	S#	The SunOS release 5.5, ABI, SVID, and SVR4 versions include <code><unistd.h></code> while the SunOS release 4.x version does not. The following <code>errno</code> flag is valid for the SunOS release 4.x version of this system call but is not valid in SunOS release 5.5, ABI, SVID, and SVR4 versions: <code>ENOTEMPTY</code> . SunOS release 5.5, ABI, SVID, and SVR4 versions set <code>errno</code> to flag <code>EEXIST</code> instead.	S#	S#	S#	N
sbrk(2)	S		N	N	S	N
select(2)	S	Now <code>select(3C)</code> .	N	N	S	N
semctl(2)	S		S	S	S	N

≡ B

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Re-lease 5.5	Notes	ABI	SVID	SVR4	BSD
semget(2)	S		S	S	S	N
semop(2)	S		S	S	S	N
send(2)	S#	Now send(3N). The following <code>errno</code> flag is valid for SunOS release 4.x send(2) system calls but is not valid in the SVR4 and SunOS 5.5 version: <code>ENOBUFS</code> .	N	N	S#	N
sendmsg(2)	S#	Now sendmsg(3N). The following <code>errno</code> flag is valid for SunOS release 4.x sendmsg(2) system calls but is not valid in the SVR4 and SunOS 5.5 version: <code>ENOBUFS</code> .	N	N	S#	N
sendto(2)	S#	Now sendto(3N). The following <code>errno</code> flag is valid for SunOS release 4.x sendto(2) system calls but is not valid in the SVR4 and SunOS 5.5 version: <code>ENOBUFS</code> .	N	N	S#	N
setaudit(2)	N		N	N	N	N
setaudit(2)	N		N	N	N	N
setdomainname(2)	A	The <code>sysinfo(2)</code> system call provides similar functionality.	N	N	N	N
setgroups(2V)	C	Int SunOS release 4.x, the <code>gidset</code> argument is of type <code>int</code> , while in SunOS release 5.5, ABI, SVID, and SVR4, the <code>grouplist</code> argument is of type <code>gid_t</code> . Also, SunOS release 5.5, ABI, SVID, and SVR4 versions include <code><unistd.h></code> and <code><sys/types.h></code> while the SunOS release 4.x version does not.	C	C	C	N
setgroups(2V) — SysV	S		S	S	S	N
sethostname(2)	A	Now sethostname(3C). The <code>sysinfo(2)</code> system call with the <code>command</code> argument set to <code>SI_SET_HOSTNAME</code> provides similar functionality.	N	N	N	S
setitimer(2)	S		N	S	S	N
setpgid(2V) — SysV	S		S	S	S	N

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Re-lease 5.5	Notes	ABI	SVID	SVR4	BSD
setpgrp(2V)	C#	The SunOS release 4.x version of <code>setpgrp()</code> has arguments <i>pid</i> and <i>pgrp</i> , and <code>setpgrp()</code> sets the process group to <i>pgrp</i> of the process indicated by <i>pid</i> . The SunOS 5.5 version of <code>setpgrp()</code> does not accept an argument, and <code>setpgrp()</code> also creates a new session. However, if <i>pgrp</i> is zero and <i>pid</i> refers to the calling process, then SunOS release 4.x <code>setpgrp()</code> call is identical to a SunOS 5.5 <code>setpgrp()</code> call with no arguments. Also, SunOS release 4.x <code>setpgrp()</code> returns a value of type <code>int</code> , while SunOS release 5.5, <code>setpgrp()</code> returns a value of type <code>pid_t</code> . Also, the SunOS 5.5 version includes <code><unistd.h></code> and <code><sys/types.h></code> while the SunOS release 4.x version does not. In SunOS release 5.5, the function <code>stashed()</code> should be used instead of <code>setpgrp()</code> , which is being phased out. It behaves identically. The following <code>errno</code> flags are valid for SunOS release 4.x <code>setpgrp()</code> system call but are not valid in SunOS release 5.5, ABI, SVID, and SVR4 versions: <code>EACCES</code> , <code>EINVAL</code> , <code>ESRCH</code> .	C#	C#	C#	N
setpgrp(2V) — SysV	S#	In SunOS release 5.5, ABI, SVID, and SVR4, the function <code>setsid()</code> should be used instead of <code>setpgrp()</code> , which is being phased out. It behaves identically. The following <code>errno</code> flags are valid for SunOS release 4.x <code>setpgrp(2V)</code> system call but is not valid in SunOS release 5.5, ABI, SVID, and SVR4 versions: <code>EACCES</code> , <code>EINVAL</code> , <code>ESRCH</code> .	S#	S#	S#	N
setpriority(2)	A	The <code>priocntl(2)</code> system call provides similar functionality.	A	A	A	S
setregid(2)	S		N	N	N	C
setreuid(2)	S		N	N	N	C
setrlimit(2)	C	Now <code>setrlimit(3C)</code> .	C	C	C	N
setsid(2V) — SysV	S		S	S	S	N
setsockopt(2)	S	Now <code>setsockopt(3N)</code> .	N	N	S	N
settimeofday(2)	S	Now <code>settimeofday(3C)</code> .	N	S	S	S
setuseraudit(2)	N		N	N	N	N

≡ B

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Re-lease 5.5	Notes	ABI	SVID	SVR4	BSD
sgetl(2)	S	Now xdr_simple(3N).	N	S	S	N
shmat(2)	S		S	S	S	N
shmctl(2)	S		S	S	S	N
shmdt(2)	S		S	S	S	N
shmget(2)	S		S	S	S	N
shutdown(2)	S	Now shutdown(3N).	N	N	S	N
sigblock(2)	A	The sigprocmask(2) system call with the <i>how</i> argument set to SIG_BLOCK provides similar functionality.	A	A	A	S
sigaction(2)	C	There is a flag in the Solaris 2.5 version, SA_RESTART, which allows a function that is interrupted by the execution of this signal's handler to be transparently restarted by the system.	N	C	C	S
sigmask(2)	A	The sigsetops(3C) routines provide similar functionality in SunOS release 5.5, ABI, SVID, and SVR4.	A	A	A	S
sigpause(2V) — SysV	S	SunOS release 4.x sigpause() system call assigns its argument (<i>sigmask</i>) to the set of masked signals while the ABI and SVID versions of sigpause remove its argument (<i>sig</i>) from the calling process's signal mask. The SVR4 and SunOS 5.5 sigpause() is compatible with SunOS release 4.x sigpause(2).	C	C	S	S
sigpending(2V) — SysV	S		S	S	S	N
sigprocmask(2V) — SysV	S		S	S	S	N
sigsetmask(2)	A	The sigprocmask(2) routine with the <i>how</i> argument set to SIG_SETMASK provides similar functionality in SunOS release 5.5, ABI, SVID, and SVR4.	A	A	A	S
sigstack(2)	A	The sigaltstack(2) system call provides similar functionality.	A	A	A	S
sigsuspend(2V) — SysV	S		S	S	S	N

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Re-lease 5.5	Notes	ABI	SVID	SVR4	BSD
sigvec(2)	A	The sigaction(2) system call provides similar functionality.	A	A	A	S
socket(2)	C#	Now socket(3N). In SunOS release 4.x, PF_IMPIPKNK is a supported <i>domain</i> , while in SVR4 and SunOS 5.5 PF_IMPIPKNK is not supported. The following <i>errno</i> flags are valid for SunOS release 4.x socket() system call but are not valid in the SVR4 and SunOS 5.5 version: ENOBUFS, EPROTOTYPE.	N	N	C#	N
socketpair(2)	S	Now socketpair(3N).	N	N	S	N
sput1(2)	S	Now xdr_simple(3N).	N	S	S	N
stat(2V) — SysV	S		S	S	S	N
statfs(2)	A	The statvfs(2) system call provides similar functionality.	A	A	A	N
swapon(2)	A	The swapctl(2) system call provides similar functionality.	N	N	N	N
symlink(2)	S		S	S	S	N
sync(2)	S		S	S	S	N
syscall(2)	N		N	N	N	S
sysconf(2V) — SysV	S	Now sysconf(3C).	S	S	S	N
tell(2V) — SysV	S		S	N	S	N
truncate(2)	S	Now truncate(3C).	N	N	S	N
umask(2V) — SysV	C	The following symbolic access modes (<sys/stat.h>) are supported by the SunOS release 4.x version of umask(), but not by SunOS release 5.5, ABI, SVID, and SVR4 versions: S_IREAD (00400), S_IWRITE (00200), S_IEXEC (00100). However, the equivalent SunOS release 5.5, ABI, SVID, and SVR4 symbolic access modes, S_IRUSR (00400), S_IWUSR (00200), S_IXUSR (00100), do have the same definitions, are defined in SunOS release 4.x <sys/stat.h>, and thus should be used.	C	C	C	N
umount(2V) — SysV	S		S	S	S	N

≡ B

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Re-lease 5.5	Notes	ABI	SVID	SVR4	BSD
uname(2V) — SysV	S		S	S	S	N
unlink(2V) — SysV	S		S	S	S	N
umount(2)	A	The <code>umount(2)</code> system call provides similar functionality.	A	A	A	N
ustat(2)	S		S	S	S	N
utimes(2)	S		N	N	N	N
vadvise(2)	N		N	N	N	N
vfork(2)	S		N	N	S	N
vhangup(2)	S		N	N	N	N
wait(2V)	C	In SunOS release 4.x, <code>wait()</code> returns a value of type <code>int</code> . In SunOS release 5.5, ABI, SVID, and SVR4, <code>wait()</code> returns a value of type <code>pid_t</code> . Also, SunOS release 5.5, ABI, SVID, and SVR4 versions include <code><sys/types.h></code> while the SunOS release 4.x version does not. The <i>union wait</i> , supported in SunOS release 4.x for backwards compatibility with previous SunOS releases, is not supported in SunOS release 5.5, ABI, SVID, and SVR4 versions. In SunOS release 4.x, <code>wait()</code> is automatically restarted when a process receives a signal while awaiting termination, unless the <code>SV_INTERRUPT</code> bit is set in the flags for that signal. In SunOS release 5.5, ABI, SVID, and SVR4, the <code>wait()</code> system call returns prematurely if a signal is received.	C	C	C	N
wait(2V) — SysV	C	The <i>union wait</i> , supported in SunOS release 4.x for backwards compatibility with previous SunOS releases, is not supported in SunOS release 5.5, ABI, SVID, and SVR4 versions. In SunOS release 4.x, <code>wait(2V)</code> is automatically restarted when a process receives a signal while awaiting termination, unless the <code>SV_INTERRUPT</code> bit is set in the flags for that signal. In SunOS release 5.5, ABI, SVID, and SVR4, the <code>wait(2)</code> function will return prematurely if a signal is received.	C	C	C	N
wait3(2V)	A	Now <code>wait3(3C)</code> . The <code>wait(2)</code> and <code>waitpid(2)</code> system calls provide similar functionality.	A	A	A	S

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Re-lease 5.5	Notes	ABI	SVID	SVR4	BSD
wait4(2V)	A	Now wait4(3C). The wait(2) and waitpid(2) system calls provide similar functionality.	A	A	A	S
waitpid(2V)	C	In SunOS release 4.x, waitpid() returns a value of type <code>int</code> . In SunOS release 5.5, ABI, SVID, and SVR4, waitpid() returns a value of type <code>pid_t</code> . The <code>pid</code> argument to waitpid() is of type <code>int</code> in SunOS release 4.x and of type <code>pid_t</code> in SunOS release 5.5, ABI, SVID, and SVR4. Also, SunOS release 5.5, ABI, SVID, and SVR4 versions include <code><sys/types.h></code> while the SunOS release 4.x version does not. The <i>union wait</i> , supported in SunOS release 4.x for backwards compatibility with previous SunOS releases, is not supported in SunOS release 5.5, ABI, SVID, and SVR4 versions. In SunOS release 4.x, waitpid() is automatically restarted when a process receives a signal while awaiting termination, unless the <code>SV_INTERRUPT</code> bit is set in the flags for that signal. In SunOS release 5.5, ABI, SVID, and SVR4, the waitpid() system call returns prematurely if a signal is received.	C	C	C	N
waitpid(2V) — SysV	C	The <i>union wait</i> , supported in SunOS release 4.x for backwards compatibility with previous SunOS releases, is not supported in SunOS release 5.5, ABI, SVID, and SVR4 versions. In SunOS release 4.x, waitpid(2V) is automatically restarted when a process receives a signal while awaiting termination, unless the <code>SV_INTERRUPT</code> bit is set in the flags for that signal. In SunOS release 5.5, ABI, SVID, and SVR4, the waitpid(2) function will return prematurely if a signal is received.	C	C	C	N

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Re-lease 5.5	Notes	ABI	SVID	SVR4	BSD
write(2V)	C#	<p>In SunOS release 4.x, if the object which descriptor refers to is marked for non-blocking I/O, using the <code>FIONBIO</code> request to <code>ioctl()</code>, or by using <code>fcntl()</code> to set the <code>FNDELAY</code> or <code>O_NDELAY</code> flag, <code>write()</code> returns <code>-1</code> and sets <code>errno</code> to <code>EWOULDBLOCK</code>.</p> <p>In SunOS release 5.5, on a <code>write()</code> to a regular file, if <code>O_NDELAY</code> or <code>O_NONBLOCK</code> is set, <code>write()</code> returns <code>-1</code> and sets <code>errno</code> to <code>EAGAIN</code>.</p> <p>On <code>write()</code> requests to a pipe or FIFO with <code>O_NONBLOCK</code> or <code>O_NDELAY</code> set, <code>write()</code> does not block the process. If some data can be written without blocking the process, <code>write()</code> writes what it can and returns the number of bytes written; otherwise, when <code>O_NONBLOCK</code> is set, it returns <code>-1</code> and sets <code>errno</code> to <code>EAGAIN</code> and when <code>O_NDELAY</code> is set, it returns <code>0</code>.</p> <p>With <code>O_NDELAY</code> set, <code>write()</code> requests for <code>{PIPE_BUF}</code> or fewer bytes either succeed completely and return <code>nbytes</code>, or return <code>0</code>. A <code>write()</code> request for greater than <code>{PIPE_BUF}</code> bytes either transfers what it can and returns the number of bytes written, or transfers no data and returns <code>0</code>. Also, if a request is greater than <code>{PIPE_BUF}</code> bytes and all data previously written to the pipe has been read, <code>write()</code> transfers at least <code>{PIPE_BUF}</code> bytes.</p> <p>The SunOS 5.5 <code>write()</code> routine does not support 4.2 BSD style non-blocking I/O.</p> <p>The following <code>errno</code> flag is valid for the SunOS release 4.x version of this system call but is not valid in the SunOS 5.5 version: <code>EWOULDBLOCK</code>.</p>	C#	C#	C#	N
write(2V) — SysV	C#	<p>The SunOS release 5.5, ABI, SVID, and SVR4 versions of <code>write()</code> does not support 4.2 BSD style non-blocking I/O.</p> <p>The following <code>errno</code> flag is valid for the SunOS release 4.x version of this system call but is not valid in SunOS release 5.5, ABI, SVID, and SVR4 versions: <code>EWOULDBLOCK</code>.</p>	C#	C#	C#	N

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Re- lease 5.5	Notes	ABI	SVID	SVR4	BSD
writev(2V)	C#	SunOS release 5.5, ABI, SVID, and SVR4 versions of writev() does not support 4.2 BSD style non-blocking I/O. The following <code>errno</code> flag is valid for the SunOS release 4.x version of this system call but is not valid in the SunOS release 5.5 release, or the ABI, SVID, or SVR4 version: EWOULDBLOCK.	C#	C#	C#	N

Library Routines Reference Table



This appendix contains the Library Routine reference table. This table lists all SunOS release 4.x library routines and shows their status in the following environments: Solaris 2.5, the ABI, the SVID, SVR4, and the SunOS/BSD Source Compatibility Package.

Using the Reference Table

- If an interface is listed as “changed” (C), a brief description of differences between the SunOS release 4.x command and the Solaris 2.5 command is provided.
- If an interface is listed as “the same” (S), the Solaris 2.5 interface supports all features of the SunOS release 4.x interface. In some cases the interface has been enhanced, but can be considered a complete superset of the SunOS release 4.x interface.
- If an interface has an “alternative” (A), check the Notes section for its replacement.
- If an interface is listed as “not available” (N), check the Notes section for information about its replacement. Routines listed in the SunOS release 5.5 column replace the SunOS release 4.x interface.

SunOS release 4.x offers a System V Software installation option that provides System V compatible versions of many routines. The System V interfaces are included in the following tables. When referring to the System V version of a SunOS release 4.x interface, the string ‘SysV’ is appended to the interface.

Routines that exist in both `/usr/lib` and `/usr/5lib` have two table entries, the first documents the `/usr/lib` routine, and the second entry documents the `/usr/5lib` routine.

For complete information on all Solaris 2.5 interfaces, see the *man Pages(3): Library Routines*.

Examples

Below are sample table entries followed by an interpretation of the entry..

SunOS release 4.x Command	SunOS release 5.5	Alternative Available and Notes	ABI	SVID	SVR4	BSD
<code>clntraw_create(3N)</code>	S	This routine is still available, but is superseded by <code>clnt_raw_create(3N)</code> in SunOS release 5.5 and SVR4.	A	A	S	N

The `clntraw_create()` routine exists in this release, but it also has a replacement routine — `clnt_raw_create()`. Applications that use `clntraw_create()` will continue to work in this release and on other SVR4 compliant systems, but these applications should be updated to use `clnt_raw_create()`. `clntraw_create()` is considered obsolete, and may not be available in future releases. If you want your application to be ABI or SVID compliant, use `clnt_raw_create()`.

SunOS release 4.x Command	SunOS release 5.5	Alternative Available and Notes	ABI	SVID	SVR4	BSD
<code>putpwent(3)</code>	S		S	S	S	N

The SunOS release 4.x `putpwent()` routine and the SunOS release 5.5 routine are the same. Applications that use this application will behave as they did in the SunOS release 4.x.

Library Routines

Table C-1 Library Routines Reference Table

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
<code>_crypt(3)</code>	<code>crypt()</code>	The <code>crypt(3C)</code> routine provides similar functionality.	N	A	A	N
<code>_longjmp(3)</code>	<code>siglongjmp()</code>	The <code>siglongjmp(3)</code> routine provides similar functionality.	A	A	A	S
<code>_setjmp(3)</code>	<code>sigsetjmp()</code>	The <code>sigsetjmp(3)</code> routine provides the same functionality when the <i>savemask</i> argument is zero. This saves the calling process's registers and stack environment, but not its <i>signalmask</i> .	A	A	A	S
<code>_tolower(3V) -- SysV</code>	S		S	S	S	N
<code>_toupper(3V) -- SysV</code>	S		S	S	S	N
<code>CHECK(3L)</code>	N		N	N	N	N
<code>HUGE(3M)</code>	C	In the SunOS release 4.x, <code>HUGE</code> is defined in <code><math.h></code> as <code>infinity(3M)</code> which produces IEEE Infinity. In SunOS release 5.5, or the SVID or SVR4 <code>HUGE</code> is defined in <code><math.h></code> as a machine dependent constant.	N	C	C	N
<code>HUGE_VAL(3M)</code>	C	In the SunOS release 4.x, <code>HUGE_VAL</code> is defined in <code><math.h></code> as <code>infinity(3M)</code> which produces IEEE Infinity. In SunOS release 5.5, or the SVID or SVR4 <code>HUGE_VAL</code> is defined in <code><math.h></code> as a machine dependent constant.	N	C	C	N
<code>MONITOR(3L)</code>	N		N	N	N	N
<code>MSG_RECVALL(3L)</code>	N		N	N	N	N
<code>SAMECV(3L)</code>	N		N	N	N	N
<code>SAMEMON(3L)</code>	N		N	N	N	N
<code>SAMETHREAD(3L)</code>	N		N	N	N	N
<code>STKTOP(3L)</code>	N		N	N	N	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
a64l(3)	S		S	S	S	N
abort(3)	S		S	S	S	N
abs(3)	S		S	S	S	N
acos(3M)	C	In the SunOS release 4.x, if the absolute value of the argument of <code>acos()</code> is greater than one, NaN is returned with an EDOM error and a DOMAIN math err. SunOS release 5.5, the SVID, or SVR4 versions return zero with an EDOM error and a DOMAIN math err.	N	C	C	N
acosh(3M)	S		N	S	S	N
addch(3V) -- SysV	S		N	S	S	N
addexportent(3)	A	The <code>/etc/dfs/sharetab</code> file replaces <code>/etc/exports</code> in SunOS release 5.5. Refer to <code>share(1M)</code> , <code>unshare(1M)</code> , and <code>sharetab(4)</code> for more information.	N	N	N	N
addmntent(3)	<code>putmntent()</code>	The <code>putmntent()</code> routine provides similar functionality. Refer to <code>getmntent(3C)</code> .	N	N	N	N
addstr(3V) -- SysV	S		N	S	S	N
agt_create(3L)	N		N	N	N	N
agt_enumerate(3L)	N		N	N	N	N
agt_trap(3L)	N		N	N	N	N
aint(3M)	N		N	N	N	N
aiocancel(3)	S		N	N	N	N
aioread(3)	S		N	N	N	N
aiowait(3)	S		N	N	N	N
aiowrite(3)	S		N	N	N	N
alarm(3V)	S		S	S	S	N
alloca(3)	S		N	N	N	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
alphasort(3)	N		N	N	N	S
anint(3M)	N		N	N	N	N
annuity(3M)	N		N	N	N	N
arc(3X)	S		N	N	N	N
asctime(3V)	C	See ctime(3V).	C	C	C	N
asin(3M)	C	In the SunOS release 4.x, if the absolute value of the argument of asin() is greater than one, NaN is returned with an EDOM error and a DOMAIN math err. SunOS release 5.5, the SVID, or SVR4 versions return zero with an EDOM error and a DOMAIN math err.	N	C	C	N
asinh(3M)	S		N	S	S	N
assert(3V)	C	The SunOS 4.x version of assert() calls exit(3C) while SunOS release 5.5, or the ABI, SVID, or SVR4 version calls abort(3C).	C	C	C	N
assert(3V) -- SysV	S		S	S	S	N
atan(3M)	S		N	S	S	N
atan2(3M)	C	In SunOS release 5.5, the SVID, or SVR4 version, atan2(0.0,0.0) returns zero and sets errno to EDOM. In the SunOS 4.x version, the same call might return +/-0.0 or +/- PI in conformance with 4.3BSD in the spirit of ANSI/IEEEStd754- 1985.	N	C	C	N
atanh(3M)	S		N	S	S	N
atof(3)	C	See strtod(3).	C	C	C	N
atoi(3)	S		S	S	S	N
atol(3)	S		S	S	S	N
atrtroff(3V) -- SysV	S		N	S	S	N
atron(3V) -- SysV	S		N	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
attrset(3V) -- SysV	S		N	S	S	N
audit_args(3)	N		N	N	N	N
audit_text(3)	N		N	N	N	N
authdes_create(3N)	authdes _seccreate()	This routine is still available, but is superceded by authdes_seccreate(3N) in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	A	N
authdes_getucred(3N)	S		S	S	S	N
auth_destroy(3N)	S		S	S	S	N
authnone_create(3N)	S		S	S	S	N
authunix_create(3N)	authsys _seccreate()	This routine is still available, but is superceded by authsys_seccreate(3N) in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	A	N
authunix_create_default(3N)	authsys _create_default()	This routine is still available, but is superceded by authsys_create_default(3N) in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	A	N
baudrate(3V) -- SysV	S		N	S	S	N
bcmp(3)	S		A	A	A	S
bcopy(3)	S		A	A	A	S
beep(3V) -- SysV	S		N	S	S	N
bindresvport(3N)	S		N	N	S	N
bootparam(3R)	S		N	N	N	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
box(3V)	C	The SunOS 4.x version of <code>box()</code> is a function while the SunOS release 5.5, or the SVID or SVR4 version of <code>box(win, verch, horch)</code> is a macro that calls <code>wborder(win, verch, verch, horch, horch, 0, 0, 0, 0)</code> . Default values defined in <code><curses.h></code> in SunOS release 5.5, or the SVID or SVR4: <code>ACS_ULCORNER</code> , <code>ACS_URCORNER</code> , <code>ACS_BLCORNER</code> , and <code>ACS_BRCORNER</code> , are used to draw the upper left and right and bottom left and right corners of the box around the window. Also, the type of arguments <code>verch</code> and <code>horch</code> in SunOS 4.x is <code>char</code> , while in SunOS release 5.5, or the SVID or SVR4 they are <code>ch</code> type.	N	C	C	S
box(3V) -- SysV	S		N	S	S	N
bsearch(3)	S		S	S	S	N
byteorder(3N)	S		N	N	S	N
bzero(3)	S		A	A	A	S
calloc(3)	S		S	S	S	N
callrpc(3N)	A	This routine is still available, but is superseded by <code>rpc_call(3N)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	N	N	S	N
catclose(3C)	S		S	S	S	N
catgetmsg(3C)	A	In SunOS release 5.5, or the ABI, SVID, or SVR4, use <code>catgets(3C)</code> followed by <code>strncpy(3)</code> to copy the catalog message from the internal buffer area to a program buffer.	A	A	A	N
catgets(3C)	S		S	S	S	N
catopen(3C)	S		S	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
cbc_crypt(3)	S		N	N	N	N
cbreak(3V) -- SysV	S		N	S	S	S
cbrt(3M)	S		N	S	S	N
ceil(3M)	S		N	S	S	N
cfgetispeed(3V)	S		S	S	S	N
cfgetospeed(3V)	S		S	S	S	N
cfree(3)	A	This routine is replaced by <code>void free(void*ptr)</code> (refer to <code>malloc(3C)</code>) in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	A	N
cfsetispeed(3V)	S		S	S	S	N
cfsetospeed(3V)	S		S	S	S	N
circle(3X)	S		N	N	N	N
clear(3V) -- SysV	S		N	S	S	S
clearerr(3V) -- SysV	S		S	S	S	N
clearok(3V) -- SysV	S		N	S	S	S
clnt_broadcast(3N)	S	This routine is still available, but is superseded by <code>rpc_broadcast(3N)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	A	N
clnt_call(3N)	S		S	S	S	N
clnt_control(3N)	S		S	S	S	N
clnt_create(3N)	S		S	S	S	N
clnt_destroy(3N)	S		S	S	S	N
clnt_freeres(3N)	S		S	S	S	N
clnt_geterr(3N)	S		S	S	S	N
clnt_pcreateerror(3N)	S		S	S	S	N
clnt_perrno(3N)	S		S	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
clnt_perror(3N)	S		S	S	S	N
clnt_spcreateerror(3N)	S		S	S	S	N
clnt_sperrno(3N)	S		S	S	S	N
clnt_sperror(3N)	S		S	S	S	N
clntraw_create(3N)	S	This routine is still available, but is superseded by <code>clnt_raw_create(3N)</code> in SunOS release 5.5 and SVR4.	A	A	S	N
clnttcp_create(3N)	S	This routine is still available, but is superseded by <code>clnt_create(3N)</code> , <code>clnt_tli_create(3N)</code> , and <code>clnt_vc_create(3N)</code> routines in SunOS release 5.5, or the ABI, SVID, or SVR4.	N	N	S	S
clntudp_bufcreate(3N)	S	This routine is still available, but is superseded by <code>clnt_create(3N)</code> , <code>clnt_tli_create(3N)</code> , and <code>clnt_dg_create(3N)</code> routines in SunOS release 5.5, or the ABI, SVID, or SVR4.	N	N	S	
clntudp_create(3N)	S	This routine is still available, but is superseded by <code>clnt_create(3N)</code> , <code>clnt_tli_create(3N)</code> , and <code>clnt_dg_create(3N)</code> routines in SunOS release 5.5, or the ABI, SVID, or SVR4.	N	N	S	S
clock(3C)	S		S	S	S	N
closedir(3V)	S		S	S	S	N
closedir(3V) -- SysV	S		S	S	S	N
closelog(3)	S		N	N	S	N
closepl(3X)	S		N	N	N	N
clrtobot(3V) -- SysV	S		N	S	S	S
clrtoeol(3V) -- SysV	S		N	S	S	S
compound(3M)	N		N	N	N	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
cont(3X)	S		N	N	N	N
copysign(3M)	N		N	N	S	N
copywin(3V) -- SysV	S		N	S	S	N
cos(3M)	C	For arguments that are much lower than zero, SunOS release 5.5, or the SVID or SVR4 version of these routines return zero because of the loss of significance. In this case, a message indicating TLOSS (see <code>matherr(3M)</code>) appears on the standard output. For cases of partial loss of significance, a PLOSS error is generated, but no error is printed. In both cases, <code>errno</code> is set to <code>ERANGE</code> . In the SunOS 4.x version, an argument reduction takes place for values exceeding $\pi/4$ in magnitude. The reduction could happen in software or hardware. The variable <code>fp_pi</code> defined in <code><math.h></code> allows changing of the precision at run time. The error exceptions occur in the IEEE 754 spirit for both versions.	N	C	C	N
cosh(3M)	S		N	S	S	N
crmode(3X)	A	This routine is replaced by <code>cbreak()</code> (see <code>cursor_inopts(3X)</code>) in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	A	N
crypt(3)	C	In the SunOS 4.x version, the first two characters of the <code>salt</code> argument are interpreted and checked for (<code>##and#S</code>) as special cases in order to call additional authentication routines (<code>pwdauth(3)</code> and <code>grpauth(3)</code> respectively). If these functions return <code>TRUE</code> , the <code>salt</code> is returned from <code>crypt</code> . Otherwise, <code>NULL</code> is returned. In SunOS release 5.5, or the SVID or SVR4 version, this functionality is not supported.	N	C	C	N
ctermid(3V) -- SysV	S		S	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
ctime(3V)	C	<p>The SunOS 4.x <code>tm</code> structure contains two fields not present in SunOS release 5.5, or the ABI, SVID, or SVR4 <code>tm</code> structure: <code>tm_zone</code> and <code>tm_gmtoff</code>. Instead SunOS release 5.5, or the ABI, SVID, or SVR4 version uses the external variable <code>timezone</code> to contain the difference (in seconds) between GMT and local standard time, and the external variable <code>daylight</code> to indicate if daylight savings should be applied. Additionally, SunOS release 5.5, or the ABI, SVID, or SVR4 version uses an external variable <code>tzname</code> to store standard and summer time zone names. These external variables (<code>timezone</code>, <code>daylight</code>, and <code>tzname</code>) are supported by the SunOS 4.x System V <code>ctime(3V)</code> library routines.</p> <p>The use of the environmental variable <code>TZ</code> differs between SunOS 4.x and SunOS release 5.5, or the ABI, SVID, or SVR4 versions. In the SunOS release 4.x, <code>TZ</code> contains the pathname of <code>tzfile-format</code> file from which to read the time conversion information. In SunOS release 5.5, or the ABI, SVID, or SVR4, <code>TZ</code> itself contains the time conversion information (of different format than the <code>tzfile-format</code>).</p>	C	C	C	N
curs_set(3V) -- SysV	S		N	S	S	N
cuserid(3V)	S		S	S	S	N
cv_broadcast(3L)	N		N	N	N	N
cv_create(3L)	N		N	N	N	N
cv_destroy(3L)	N		N	N	N	N
cv_enumerate(3L)	N		N	N	N	N
cv_notify(3L)	N		N	N	N	N
cv_send(3L)	N		N	N	N	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
cv_wait(3L)	N		N	N	N	N
cv_waiters(3L)	N		N	N	N	N
dbm_clearerr(3)	S		N	N	N	N
dbm_close(3X)	S	The dbm_close(3) routine provides similar functionality.	N	N	N	N
dbm_delete(3)	S	The dbm_delete(3) routine provides similar functionality.	N	N	N	N
dbm_error(3)	S		N	N	N	N
dbm_fetch(3)	S		N	N	N	N
dbm_firstkey(3)	S		N	N	N	N
dbm_nextkey(3)	S		N	N	N	N
dbm_open(3)	S		N	N	N	N
dbm_store(3)	S		N	N	N	N
dbmclose(3X)	N		N	N	N	S
dbminit(3X)	S		N	N	N	S
decimal_to_double(3)	S		N	N	N	N
decimal_to_extended(3)	S		N	N	N	N
decimal_to_floating(3)	S		N	N	N	N
decimal_to_single(3)	S		N	N	N	N
def_prog_mode(3V) -- SysV	S		N	S	S	N
def_shell_mode(3V) -- SysV	S		N	S	S	N
del_curterm(3V) -- SysV	S		N	S	S	N
delay_output(3V) -- SysV	S		N	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
delch(3V) -- SysV	S		N	S	S	S
delete(3X)	A		N	N	N	S
deleteln(3V) -- SysV	S		N	S	S	S
des_crypt(3)	N		N	N	N	N
des_setparity(3)	S		N	N	N	N
delwin(3V) -- SysV	S		N	S	S	S
dlclose(3X)	S		N	N	S	N
dLError(3X)	S		N	N	S	N
dlopen(3X)	S		N	N	S	N
dlsym(3X)	S		N	N	S	N
dn_comp(3)	S		N	N	S	N
dn_expand(3)	S		N	N	S	N
double_to_decimal(3)	S		N	N	N	N
douupdate(3V) -- SysV	S		N	S	S	N
draino(3V) -- SysV	S		N	N	N	N
drand48(3)	S		N	S	S	N
dysize(3V)	N		N	N	N	N
ecb_crypt(3)	S		N	N	N	N
echo(3V) -- SysV	S		N	S	S	S
echochar(3V) -- SysV	S		N	S	S	N
econvert(3)	S		N	N	N	N
ecvt(3)	S		N	N	S	N
edata(3)	S		N	N	S	N
encrypt(3)	S		N	S	S	N
end(3)	S		N	N	S	N
endac(3)	N		N	N	N	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
endexportent(3)	A	The <code>/etc/dfs/sharetab</code> file replaces <code>/etc/exports</code> in SunOS release 5.5, or the ABI, SVID, or SVR4. Refer to <code>share(1M)</code> , <code>unshare(1M)</code> , and <code>sharetab(4)</code> for more information.	A	A	A	N
endfsent(3)	A	This routine is replaced by <code>fclose(3)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	A	N
endgraent(3)	N		N	N	N	N
endgrent(3V)	S		S	S	S	N
endhostent(3N)	S		N	N	S	N
endmntent(3)	A	This routine is replaced by <code>fclose(3)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	A	N
endnetent(3N)	S		N	N	S	N
endnetgrent(3N)	S		N	N	N	N
endprotoent(3N)	S		N	N	S	N
endpwaent(3)	N		N	N	N	N
endpwent(3V)	S		S	S	S	N
endrpcent(3N)	S		N	N	S	N
endservent(3N)	S		N	N	S	N
endttyent(3)	N	Refer to <code>ttymon(1)</code> and <code>ttydefs(4)</code> for information about SunOS release 5.5 tty system.	N	N	N	N
endusershell(3)	S		N	N	N	N
endwin(3V)	C	The SunOS 4.x version of <code>endwin()</code> return value is undefined, while SunOS release 5.5, or the SVID or SVR4 version returns OK upon success, otherwise it returns ERR.	N	C	C	S
endwin(3V) -- SysV	S		N	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
erand48(3)	S		N	S	S	N
erase(3V) -- SysV	S		N	S	S	S
erasechar(3V) -- SysV	S		N	S	S	N
erf(3M)	S		N	S	S	N
erfc(3M)	S		N	S	S	N
errno(3)	S		N	N	N	N
etext(3)	S		N	N	S	N
ether(3R)	N		N	N	N	N
ether_aton(3N)	S		N	N	S	N
ether_hostton(3N)	S		N	N	S	N
ether_line(3N)	S		N	N	S	N
ether_ntoa(3N)	S		N	N	S	N
ether_ntohost(3N)	S		N	N	S	N
exc_bound(3L)	N		N	N	N	N
exc_handle(3L)	N		N	N	N	N
exc_notify(3L)	N		N	N	N	N
exc_on_exit(3L)	N		N	N	N	N
exc_raise(3L)	N		N	N	N	N
exc_unhandle(3L)	N		N	N	N	N
exc_uniqpatt(3L)	N		N	N	N	N
execl(3V)	C		C	C	C	N
execl(3V) -- SysV	S		S	S	S	N
execle(3V)	C		C	C	C	N
execle(3V) -- SysV	S		S	S	S	N
execlp(3V)	C		C	C	C	N
execlp(3V) -- SysV	S		S	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
execv(3V)	C		C	C	C	N
execv(3V) -- SysV	S		S	S	S	N
execvp(3V)	C		C	C	C	N
execvp(3V) -- SysV	S		S	S	S	N
exit(3)	C	Both the SunOS 4.x and SunOS release 5.5, or the ABI, SVID, or SVR4 <code>exit()</code> routines do additional processing before the process exits. The SunOS 4.x <code>exit()</code> calls all functions registered by the <code>on_exit(3)</code> routine while SunOS release 5.5, or the ABI, SVID, or SVR4 <code>exit()</code> calls all functions registered by the <code>atexit()</code> routine. If no functions have been added using the <code>on_exit(3)</code> routine, then the SunOS 4.x and SunOS release 5.5, or the ABI, SVID, or SVR4 versions of <code>exit()</code> are compatible.	C	C	C	N
exp(3M)	C	In SunOS release 5.5, or the SVID or SVR4 version, <code>exp()</code> returns HUGE for overflow and 0 for underflow. In the SunOS 4.x version, the return values are IEEE overflow and underflow (implementation-defined). In the SunOS release 4.x, since HUGE is defined as +Infinity, <code>exp(HUGE)</code> and <code>exp(-HUGE)</code> do not overflow or underflow, hence no <code>errno</code> is produced. SunOS release 5.5, or the SVID or SVR4 version sets <code>errno</code> to ERANGE.	N	C	C	N
exp10(3M)	N		N	N	N	N
exp2(3M)	N		N	N	N	N
expm1(3M)	N		N	N	N	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
exportent(3)	A	The /etc/dfs/sharetab file replaces /etc/exports in SunOS release 5.5, or the ABI, SVID, or SVR4. Refer to share(1M), unshare(1M), and sharetab(4) for more information.	A	A	A	N
extended_to_decimal(3)	S		N	N	N	N
fabs(3M)	S		N	S	S	N
fclose(3S)	S		S	S	S	N
fconvert(3)	S		N	N	N	N
fcvt(3)	S		N	N	S	N
fdopen(3V)	S		S	S	S	N
feof(3V)	S		S	S	S	N
ferror(3V)	S		S	S	S	N
fetch(3X)	A	This routine is replaced by dbm_fetch(3) in SunOS release 5.5.	N	N	N	S
fflush(3S)	S		S	S	S	N
ffs(3)	S		N	N	S	N
fgetc(3V)	S		S	S	S	N
fgetgraent(3)	N		N	N	N	N
fgetgrent(3V)	S		N	S	S	N
fgetpwaent(3)	N		N	N	N	N
fgetpwent(3V)	S		N	S	S	N
fgets(3S)	S		S	S	S	N
fileno(3V)	S		S	S	S	N
file_to_decimal(3)	N		N	N	N	N
filter(3V) -- SysV	S		N	S	S	N
finite(3M)	N		N	N	N	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
firstkey(3X)	dbm _firstkey()	This routine is replaced by dbm_firstkey(3) in SunOS release 5.5.	N	N	N	S
fixterm(3V)	A	The reset_prog_mode(3X) routine provides similar functionality.	N	A	A	N
flash(3V) -- SysV	S		N	S	S	N
floatingpoint(3)	S		N	N	N	N
floor(3M)	S		N	S	S	N
flushinp(3V) -- SysV	S		N	S	S	N
flusok(3X)	N		N	N	N	S
fmod(3M)	C	In SunOS release 5.5, or the SVID or SVR4 version, fmod(x, 0.0) returns x and sets errno to EDOM. In the SunOS 4.x version, the same call returns NaN in conformance with 4.3 BSD and in the spirit of ANSI/IEEE Std 754-1985.	N	C	C	N
fopen(3V)	S		S	S	S	S
fp_class(3M)	N		N	N	N	N
fprintf(3V)	S		S	S	S	S
fputc(3S)	S		S	S	S	N
fputs(3S)	S		S	S	S	N
fread(3S)	S		S	S	S	N
free(3)	S		S	S	S	N
freopen(3V)	S		S	S	S	S
frexp(3M)	S		N	S	S	N
fscanf(3V)	S		S	S	S	N
fseek(3S)	S		S	S	S	N
ftell(3S)	S		S	S	S	N
ftime(3V)	C	Now ftime(3C).	A	A	A	S

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
ftok(3)	S		S	S	S	N
ftw(3)	S		S	S	S	N
func_to_decimal(3)	N		N	N	N	N
fwrite(3S)	S		S	S	S	N
gamma(3M)	S		N	S	S	N
garbagedlines(3V) -- SysV	S		N	N	N	N
gcd(3X)	S		N	N	N	N
gconvert(3)	S		N	N	N	N
gcvt(3)	S		N	N	S	N
getacdir(3)	N		N	N	N	N
getacflg(3)	N		N	N	N	N
getacinfo(3)	N		N	N	N	N
getacmin(3)	N		N	N	N	N
getauditflagsbin(3)	N		N	N	N	N
getauditflagschar(3)	N		N	N	N	N
getbegyx(3V) -- SysV	S		N	S	S	N
getc(3V)	S		S	S	S	N
getcap(3X)	N		N	N	N	S
getch(3V)	C	In SunOS release 5.5, or the SVID or SVR4, if the window is not a pad, and it has been moved or modified since the last call to wrefresh(), wrefresh() will be called before another character is read. In the SunOS release 4.x, wrefresh() will not be called under these circumstances.	N	C	C	S
getch(3V) -- SysV	C		N	C	C	S
getchar(3V)	S		S	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
getcwd(3V)	S	The SVR4 and SunOS 5.5 <code>getcwd()</code> routine is compatible with the SunOS 4.x <code>getcwd()</code> . In the SunOS release 4.x, if <code>buf</code> is a NULL pointer, <code>getcwd()</code> obtains <code>size</code> bytes of space using <code>malloc(3)</code> . This capability is not supported by the ABI and SVID version of <code>getcwd()</code> .	C	C	S	N
getenv(3V)	S		S	S	S	N
getexportent(3)	A	The <code>/etc/dfs/sharetab</code> file replaces <code>/etc/exports</code> in SunOS release 5.5, or the ABI, SVID, or SVR4. Refer to <code>share(1M)</code> , <code>unshare(1M)</code> , and <code>sharetab(4)</code> for more information.	A	A	A	N
getexportopt(3)	A	The <code>/etc/dfs/sharetab</code> file replaces <code>/etc/exports</code> in SunOS release 5.5, or the ABI, SVID, or SVR4. Refer to <code>share(1M)</code> , <code>unshare(1M)</code> , and <code>sharetab(4)</code> for more information.	A	A	A	N
getfauditflags(3)	N		N	N	N	N
getfsent(3)	<code>getvfsent()</code>	This routine is replaced by <code>getvfsent(3)</code> in SunOS release 5.5.	N	N	N	N
getfsfile(3)	<code>getvfsfile()</code>	This routine is replaced by <code>getvfsfile(3)</code> in SunOS release 5.5.	N	N	N	N
getfsspec(3)	<code>getvfsfile()</code>	This routine is replaced by <code>getvfsfile(3)</code> in SunOS release 5.5.	N	N	N	N
getfstype(3)	<code>getvfsany()</code>	This routine is replaced by <code>getvfsany(3)</code> in SunOS release 5.5.	N	N	N	N
getgraent(3)	N		N	N	N	N
getgranam(3)	N		N	N	N	N
getgrent(3V)	S		S	S	S	N
getgrgid(3V)	S		S	S	S	N
getgrnam(3V)	S		S	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
gethostbyaddr(3N)	S		N	N	S	N
gethostbyname(3N)	S		N	N	S	N
gethostent(3N)	S		N	N	S	N
getlogin(3V)	S		S	S	S	N
getmaxyx(3V) -- SysV	S		N	S	S	N
getmntent(3)	C	The SunOS 4.x <code>getmntent()</code> routine and SunOS release 5.5, or the SVID or SVR4 <code>getmntent()</code> routine are incompatible. The SunOS 4.x <code>getmntent()</code> returns a pointer to an object of type <code>mntent</code> while SunOS release 5.5, or the SVID or SVR4 <code>getmntent()</code> returns <code>int</code> . Additionally, SunOS release 5.5, or the SVID or SVR4 <code>getmntent()</code> uses a different incompatible structure type (<code>mnttab</code>) to return the file entry type. Additionally, null pointers are returned for corresponding '-' entries in <code>/etc/vfstab</code> .	N	C	C	N
get_myaddress(3N)	S	This routine is still available, but is superseded by <code>netdir_getbyname(3N)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	S	N	S	N
getnetbyaddr(3N)	S		N	N	S	N
getnetbyname(3N)	S		N	N	S	N
getnetent(3N)	S		N	N	S	N
getnetgrent(3N)	N		N	N	N	N
getnetname(3N)	S		S	S	S	N
getopt(3)	S		S	S	S	N
getpass(3V)	S		S	S	S	N
getprotobyname(3N)	S		N	N	S	N
getprotobynumber(3N)	S		N	N	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
getprotoent(3N)	S		N	N	S	N
getpublickey(3R)	S		S	S	S	N
getpw(3)	S		N	N	S	N
getpwaent(3)	N		N	N	N	N
getpwanam(3)	N		N	N	N	N
getpwent(3V)	S		S	S	S	N
getpwnam(3V)	S		S	S	S	N
getpwuid(3V)	S		S	S	S	N
getrpcbyname(3N)	S		S	S	S	N
getrpcbynumber(3N)	S		S	S	S	N
getrpcent(3N)	S		S	S	S	N
getrpcport(3R)	A	pmap_getport() can be used to get the same result.	N	N	N	N
gets(3S)	S		S	S	S	N
getsecretkey(3R)	S		S	S	S	N
getservbyname(3N)	S		N	N	S	N
getservbyport(3N)	S		N	N	S	N
getservent(3N)	S		N	N	S	N
getstr(3V) -- SysV	C		N	C	C	S
getsubopt(3)	S		S	S	S	N
getsyx(3V) -- SysV	S		N	S	S	N
gettext(3)	S		N	N	N	N
gettmode(3V)	C	SunOS release 5.5 header file <curses.h> automatically includes the headers <stdio.h> and <unctrl.h> and if CURS_PERFORMANCE is defined it defines most commonly used routines as macros for increased performance.	N	N	N	S

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
gettmode(3V) -- SysV	S		N	N	N	N
getttyent(3)	A	Refer to <code>ttymon(1)</code> and <code>ttydefs(4)</code> for information about SunOS release 5.5 tty system.	N	N	N	N
getttynam(3)	A	Refer to <code>ttymon(1)</code> and <code>ttydefs(4)</code> for information about SunOS release 5.5 tty system.	N	N	N	N
getusershell(3)	S		N	N	N	N
getw(3V)	S		S	S	S	N
getwd(3)	S	Now <code>getwd(3C)</code> .	A	A	A	S
getyx(3V) -- SysV	S		N	S	S	S
gmtime(3V)	C	See <code>ctime(3V)</code> .	C	C	C	N
grpauth(3)	N		N	N	N	N
gsignal(3)	S		N	N	S	N
gtty(3C)	A	The <code>termio(7)</code> interface provides similar functionality.	A	A	A	N
halfdelay(3V) -- SysV	S		N	S	S	N
has_ic(3V) -- SysV	S		N	S	S	N
has_il(3V) -- SysV	S		N	S	S	N
hasmntopt(3)	N		N	N	N	N
hcreate(3)	S		S	S	S	N
hdestroy(3)	S		S	S	S	N
host2netname(3N)	S		S	S	S	N
hsearch(3)	S		S	S	S	N
hypot(3M)	S		N	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
idlok(3V)	C	The SunOS 4.x version of <code>idlok()</code> sets an insert/delete line flag for the window which is ignored, while SunOS release 5.5, or the SVID or SVR4 version of <code>idlok()</code> sets a flag that controls whether or not the insert/delete line feature is actually used.	N	C	C	S
idlok(3V) -- SysV	S		N	S	S	N
ieee_flags(3M)	N		N	N	N	N
ieee_functions(3M)	S		N	N	N	N
ieee_handler(3M)	N		N	N	N	N
ieee_retrospective(3M)	N		N	N	N	N
ilogb(3M)	N		N	N	N	N
inch(3V) -- SysV	S		N	S	S	S
index(3)	S	Now is <code>index(3C)</code> .	A	A	A	S
inet_lnaof(3N)	S		N	N	S	N
inet_makeaddr(3N)	S		N	N	S	N
inet_netof(3N)	S		N	N	S	N
inet_network(3N)	S		N	N	S	N
inet_ntoa(3N)	S		N	N	S	N
infinity(3M)	N		N	N	N	N
initgroups(3)	S		S	S	S	N
initscr(3V)	C	The SunOS 4.x version of <code>initscr()</code> is a function while SunOS release 5.5, or the SVID or SVR4 version is a macro that calls <code>initscr32()</code> . If errors occur, the SunOS 4.x <code>initscr()</code> function returns <code>ERR</code> , while SunOS release 5.5, or the SVID or SVR4 writes an appropriate error message to the standard error and exits.	N	C	C	S

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
initscr(3V) -- SysV	S		N	S	S	N
initstate(3)	S		N	A	A	S
innetgr(3N)	S		N	N	N	N
insch(3V) -- SysV	S		N	S	S	S
insertln(3V) -- SysV	S		N	S	S	S
insque(3)	S		N	N	S	N
intrflush(3V) -- SysV	S		N	S	S	N
ipalloc(3R)	N		N	N	N	N
irint(3M)	N		N	N	N	N
isalnum(3V)	S		S	S	S	N
isalpha(3V)	S		S	S	S	N
isascii(3V)	S		S	S	S	N
isatty(3V)	S		S	S	S	N
iscntrl(3V)	S		S	S	S	N
isdigit(3V)	S		S	S	S	N
isendwin(3V) -- SysV	S		N	S	S	N
isgraph(3V)	S		S	S	S	N
isinf(3M)	N		N	N	N	N
islower(3V)	S		S	S	S	N
isnan(3M)	S		S	N	N	N
isnormal(3M)	N		N	N	N	N
isprint(3V)	S		S	S	S	N
ispunct(3V)	S		S	S	S	N
issecure(3)	N		N	N	N	N
isspace(3V)	S		S	S	S	N
issubnormal(3M)	N		N	N	N	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
isupper(3V)	S	In the SunOS release 4.x, j0(HUGE), j1(HUGE), jn(4, HUGE) will return zero with no error indication. In SunOS release 5.5, or the SVID or SVR4 these routines will return zero, set errno to ERANGE, and print a message indicating a TLOSS math error on the standard error output.	S	S	S	N
isxdigit(3V)	S		S	S	S	N
iszero(3M)	N		N	N	N	N
itom(3X)	S		N	N	N	N
j0(3M)	C		N	C	C	N
j1(3M)	C		N	C	C	N
jn(3M)	C		N	C	C	N
jrand48(3)	S		N	S	S	N
key_decryptsession(3N)	S		S	S	S	N
key_encryptsession(3N)	S		S	S	S	N
key_gendes(3N)	S		S	S	S	N
key_setsecret(3N)	S		S	S	S	N
keyname(3V) -- SysV	S		N	S	S	N
keypad(3V) -- SysV	S		N	S	S	N
killchar(3V) -- SysV	S		N	S	S	N
klm_prot(3R)	S		N	N	N	N
kvm_close(3K)	S		N	N	N	N
kvm_getcmd(3K)	S		N	N	N	N
kvm_getproc(3K)	S		N	N	N	N
kvm_getu(3K)	S		N	N	N	N
kvm_nextproc(3K)	S		N	N	N	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
kvm_nlist(3K)	S		N	N	N	N
kvm_open(3K)	S		N	N	N	N
kvm_read(3K)	S		N	N	N	N
kvm_setproc(3K)	S		N	N	S	N
kvm_write(3K)	S		N	N	N	N
l3tol(3C)	S		N	N	S	N
l64a(3)	S		S	S	S	N
label(3X)	S		N	N	N	N
lcong48(3)	S		N	S	S	N
ldaclose(3X)	N		N	N	N	N
ldahread(3X)	N		N	N	N	N
ldaopen(3X)	N		N	N	N	N
ldclose(3X)	N		N	N	N	N
ldexp(3M)	C	SunOS 4.x version of <code>ldexp()</code> differs from SunOS release 5.5, or the ABI, SVID, or SVR4 version only in the case of overflow. In the SunOS release 4.x, <code>ldexp()</code> returns (+/-) 1.0e999 if the correct value would overflow, while SunOS release 5.5, or the ABI, SVID, or SVR4 <code>ldexp()</code> returns (+/-) HUGE (according to the sign of value). Both versions set <code>errno</code> to ERANGE.	C	C	C	S
ldfcn(3)	N		N	N	N	N
ldfhread(3X)	N		N	N	N	N
ldgetname(3X)	N		N	N	N	N
ldlinit(3X)	N		N	N	N	N
ldlitem(3X)	N		N	N	N	N
ldlread(3X)	N		N	N	N	N
ldlseek(3X)	N		N	N	N	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
ldnlseek(3X)	N		N	N	N	N
ldnrseek(3X)	N		N	N	N	N
ldnshread(3X)	N		N	N	N	N
ldnsseek(3X)	N		N	N	N	N
ldohseek(3X)	N		N	N	N	N
ldopen(3X)	N		N	N	N	N
ldrseek(3X)	N		N	N	N	N
ldshread(3X)	N		N	N	N	N
ldsseek(3X)	N		N	N	N	N
ldtbindex(3X)	N		N	N	N	N
ldtbread(3X)	N		N	N	N	N
ldtbseek(3X)	N		N	N	N	N
leaveok(3V) -- SysV	S		N	S	S	S
lfind(3)	S		S	S	S	N
lgamma(3M)	S		N	S	S	N
line(3X)	S		N	N	N	N
linemod(3X)	S		N	N	N	N
localdtconv(3)	N		N	N	N	N
localeconv(3)	S		S	S	S	N
localtime(3V)	C	See ctime(3V).	C	C	C	N
lockf(3)	S		S	S	S	N
log(3M)	C	In the SunOS release 4.x, when log() produces undefined results (for example, log(-1.0)) it returns NaN, with an EDOM error and a DOMAIN matherr. In SunOS release 5.5, or the SVID or SVR4, it returns -HUGE with an EDOM error and DOMAIN matherr.	N	C	C	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
log10(3M)	C	In the SunOS release 4.x, when <code>log10()</code> produces undefined results (for example, <code>log10(0)</code>) it returns NaN, with an EDOM error and a DOMAIN matherr. In SunOS release 5.5, or the SVID or SVR4, it returns - HUGE with an EDOM error and DOMAIN matherr.	N	C	C	N
log1p(3M)	N		N	N	N	N
log2(3M)	N		N	N	N	N
logb(3M)	S		N	C	C	N
longjmp(3V)	S		S	S	S	S
longname(3V)	C	The SunOS 4.x version of <code>longname()</code> requires two arguments, <i>termbuf</i> and <i>name</i> , which do not exist in SunOS release 5.5, or the SVID or SVR4 version. <i>termbuf</i> is a pointer to the terminal entry from <code>termcap</code> , which is replaced by <code>terminfo</code> in SunOS release 5.5, or the SVID or SVR4. <i>name</i> is a pointer to a buffer to hold the result. Since both versions return the same information, simply remove the two arguments from the SunOS 4.x call to port to SunOS release 5.5, or the SVID or SVR4.	N	C	C	S
longname(3V) -- SysV	S		N	S	S	N
lrand48(3)	S		N	S	S	N
lsearch(3)	S		S	S	S	N
lto13(3C)	S		N	N	S	N
lwp_checkstkset(3L)	N		N	N	N	N
lwp_create(3L)	N		N	N	N	N
lwp_ctxinit(3L)	N		N	N	N	N
lwp_ctxmemget(3L)	N		N	N	N	N
lwp_ctxmemset(3L)	N		N	N	N	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
lwp_ctxremove(3L)	N		N	N	N	N
lwp_ctxset(3L)	N		N	N	N	N
lwp_datastk(3L)	N		N	N	N	N
lwp_destroy(3L)	N		N	N	N	N
lwp_enumerate(3L)	N		N	N	N	N
lwp_errstr(3L)	N		N	N	N	N
lwp_fpset(3L)	N		N	N	N	N
lwp_geterr(3L)	N		N	N	N	N
lwp_getregs(3L)	N		N	N	N	N
lwp_getstate(3L)	N		N	N	N	N
lwp_join(3L)	N		N	N	N	N
lwp_libcset(3L)	N		N	N	N	N
lwp_newstk(3L)	N		N	N	N	N
lwp_perror(3L)	N		N	N	N	N
lwp_ping(3L)	N		N	N	N	N
lwp_resched(3L)	N		N	N	N	N
lwp_resume(3L)	N		N	N	N	N
lwp_self(3L)	N		N	N	N	N
lwp_setpri(3L)	N		N	N	N	N
lwp_setregs(3L)	N		N	N	N	N
lwp_setstkcache(3L)	N		N	N	N	N
lwp_sleep(3L)	N		N	N	N	N
lwp_stkcswset(3L)	N		N	N	N	N
lwp_suspend(3L)	N		N	N	N	N
lwp_yield(3L)	N		N	N	N	N
madd(3X)	S		N	N	N	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
madvise(3)	S		N	N	N	N
malloc(3)	S		S	S	S	N
malloc_debug(3)	S		N	N	N	N
malloc_verify(3)	S		N	N	N	N
malloccmap(3)	S		N	N	N	N
matherr(3M)	S		N	S	S	N
max_normal(3M)	N		N	N	N	N
max_subnormal(3M)	N		N	N	N	N
mblen(3)	S		S	S	S	N
mbstowcs(3)	S		S	S	S	N
mbtowc(3)	S		S	S	S	N
mcmp(3X)	S		N	N	N	N
mdiv(3X)	S		N	N	N	N
memalign(3)	S		N	N	S	N
memccpy(3)	S		S	S	S	N
memchr(3)	S		S	S	S	N
memcmp(3)	S		S	S	S	N
memcpy(3)	S		S	S	S	N
memset(3)	S		S	S	S	N
meta(3V) -- SysV	S		N	S	S	N
mfree(3X)	S		N	N	N	N
min(3X)	S		N	N	N	N
min_normal(3M)	N		N	N	N	N
min_subnormal(3M)	N		N	N	N	N
mkstemp(3)	S	The <code>mktemp(3C)</code> routine provides similar functionality.	A	A	A	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
mktemp(3)	C	The SunOS 4.x <code>mktemp()</code> routine replaces the trailing X characters of <code>template</code> with a letter and the current process ID. SunOS release 5.5, or the ABI, SVID, or SVR4 version only specifies that it will replace the six trailing Xs with a character string that can be used to create a unique filename. If the application does not depend on the specific name of the filename (that is, the application only cares that the name is unique), the SunOS 4.x and SunOS release 5.5, or the ABI, SVID, or SVR4 versions of <code>mktemp()</code> are compatible.	C	C	C	N
mlock(3)	S		S	S	S	N
mlockall(3)	S		S	S	S	N
modf(3M)	S		N	S	S	N
mon_break(3L)	N		N	N	N	N
mon_cond_enter(3L)	N		N	N	N	N
mon_create(3L)	N		N	N	N	N
mon_destroy(3L)	N		N	N	N	N
mon_enter(3L)	N		N	N	N	N
mon_enumerate(3L)	N		N	N	N	N
mon_exit(3L)	N		N	N	N	N
mon_waiters(3L)	N		N	N	N	N
moncontrol(3)	A	This routine is replaced by <code>profil(2)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	A	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
monitor(3)	C	<p>The SunOS 4.x <code>monitor()</code> routine differs from SunOS release 5.5, or the ABI, SVID, or SVR4 version in the following respects: In the SunOS release 4.x, to profile the entire program it is sufficient to use:</p> <pre>extern etext(); monitor(N_TXTOFF(0),etext, buf, bufsize, nfunc);</pre> <p>While with SunOS release 5.5, or the ABI, SVID, or SVR4 <code>monitor()</code> routine, it is sufficient to use:</p> <pre>extern int etext(); monitor((int(*)())2,etext, buf, bufsize, nfunc);</pre> <p>In the SunOS release 4.x, to stop execution monitoring and write the results, to the <code>buf</code> defined previously use:</p> <pre>monitor(0);</pre> <p>While with SunOS release 5.5, or the ABI, SVID, or SVR4 <code>monitor</code> routine, use:</p> <pre>monitor((int(*)())0,(int(*)())0,(W ORD*) 0, 0,0);</pre> <p>The <code>prof(1)</code> command can then be used to examine the results.</p>	C	C	C	N
monstartup(3)	profil(0)	<p>This routine is replaced by <code>profil(2)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.</p>	A	A	A	N
mout(3X)	S		N	N	N	N
move(3V) -- SysV	S		N	S	S	S
mrnd48(3)	S		N	S	S	N
msg_enumrecv(3L)	N		N	N	N	N
msg_enumsend(3L)	N		N	N	N	N
msg_recv(3L)	N		N	N	N	N
msg_reply(3L)	N		N	N	N	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
msg_send(3L)	N		N	N	N	N
msub(3X)	S		N	N	N	N
msync(3)	C	The following errno flag is valid for the SunOS 4.x version of this system call but is not valid in SunOS release 5.5, or the ABI, SVID, or SVR4 version: EIO. In the SunOS 4.x version errno flag is set to EPERM if MS_INVALIDATE was specified and one or more of the pages is locked in memory, while in SunOS release 5.5, or the ABI, SVID, or SVR4 version, errno is set to EBUSY instead.	C	C	C	N
mtx(3X)	S		N	N	N	N
mult(3X)	S		N	N	N	N
munlock(3)	S		S	S	S	N
munlockall(3)	S		S	S	S	N
mvaddch(3V) -- SysV	S		N	S	S	N
mvaddstr(3V) -- SysV	S		N	S	S	N
mvcur(3V)	C	The SunOS 4.x version of mvcur() return value is undefined, while SunOS release 5.5, or the SVID or SVR4 version returns OK upon success otherwise it returns ERR.	N	C	C	S
mvcur(3V) -- SysV	S		N	S	S	N
mvdelch(3V) -- SysV	S		N	S	S	N
mvgetch(3V) -- SysV	C	In SunOS release 5.5, or the SVID or SVR4, if the window is not a pad, and it has been moved or modified since the last call to wrefresh(), wrefresh() will be called before another character is read. In the SunOS release 4.x, wrefresh() will not be called under these circumstances.	N	C	C	N
mvgetstr(3V) -- SysV	C	See getstr(3V) — Sys V.	N	C	C	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
<code>mvinch(3V)</code> -- SysV	S		N	S	S	N
<code>mvinsch(3V)</code> -- SysV	S		N	S	S	N
<code>mvprintw(3V)</code>	C	See <code>wprintw(3V)</code> .	N	C	C	S
<code>mvprintw(3V)</code> -- SysV	S		N	S	S	N
<code>mvscanw(3V)</code>	C	See <code>wscanw(3V)</code> .	N	C	C	S
<code>mvscanw(3V)</code> -- SysV	S		N	S	S	N
<code>mvwaddch(3V)</code> -- SysV	S		N	S	S	N
<code>mvwaddstr(3V)</code> -- SysV	S		N	S	S	N
<code>mvwdelch(3V)</code> -- SysV	S		N	S	S	N
<code>mvwgetch(3V)</code> -- SysV	C	In SunOS release 5.5, or the SVID or SVR4, if the window is not a pad, and it has been moved or modified since the last call to <code>wrefresh()</code> , <code>wrefresh()</code> will be called before another character is read. In the SunOS release 4.x, <code>wrefresh()</code> will not be called under these circumstances.	N	C	C	N
<code>mvwgetstr(3V)</code> -- SysV	C	See <code>getstr(3V)</code> — Sys V.	N	C	C	N
<code>mvwin(3V)</code>	C	The SunOS 4.x version of <code>mvwin()</code> can be used to move subwindows, while on SunOS release 5.5, or the SVID or SVR4, <code>mvderwin()</code> should be used to move subwindows (or derived windows) inside their parent windows.	N	C	C	S
<code>mvwin(3V)</code> -- SysV	S		N	S	S	N
<code>mvwinch(3V)</code> -- SysV	S		N	S	S	N
<code>mvwinsch(3V)</code> -- SysV	S		N	S	S	N
<code>mvwprintw(3V)</code>	C		N	C	C	S
<code>mvwprintw(3V)</code> -- SysV	S		N	S	S	N
<code>mvwscanw(3V)</code>	C	See <code>wscanw(3V)</code> .	N	C	C	S
<code>mvwscanw(3V)</code> -- SysV	S		N	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
napms(3V) -- SysV	S		N	S	S	N
net_addr(3N)	S		N	N	S	N
netname2host(3N)	S		S	S	S	N
netname2user(3N)	S		S	S	S	N
newpad(3V) -- SysV	S		N	S	S	N
newterm(3V) -- SysV	S		N	S	S	N
newwin(3V) -- SysV	S		N	S	S	S
nextafter(3M)	S		N	S	S	N
nextkey(3X)	dbm_nextkey()	This routine is replaced by dbm_nextkey(3) in SunOS release 5.5, or the ABI, SVID, or SVR4.	N	N	N	S
nice(3V)	S		S	S	S	S
nint(3M)	N		N	N	N	N
nl(3V) -- SysV	S		N	S	S	S
nl_init(3C)	N		N	N	N	N
nl_langinfo(3C)	S		S	S	S	N
nlist(3V)	C	The SunOS 4.x version of nlist() returns the number of symbols not found, or -1 on error. While SunOS release 5.5, or the SVID or SVR4 version returns 0 on success, and -1 on error. Note that SunOS release 5.5 nlist assumes an ELF format file and the 4.1 nlist() works only on a.out format files.	N	C	C	S
nlm_prot(3R)	S		N	N	N	N
nocbreak(3V) -- SysV	S		N	S	S	S
nocrmode(3X)	S		N	N	S	N
nodelay(3V) -- SysV	S		N	S	S	N
noecho(3V) -- SysV	S		N	S	S	S
nonl(3V) -- SysV	S		N	S	S	S

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
nonstandard_arithmetic(3M)	N		N	N	N	N
noraw(3V) -- SysV	S		N	S	S	S
notimeout(3V) -- SysV	S		N	S	S	N
nrand48(3)	S		N	S	S	N
ntohl(3N)	S		N	N	S	N
ntohs(3N)	S		N	N	S	N
on_exit(3)	atexit()	This routine is replaced by <code>atexit(3C)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4. Note that functions registered using <code>atexit</code> are called without arguments.	A	A	A	N
opendir(3V)	C	The DIR structure of SunOS release 5.5, or the ABI, SVID, or SVR4 does not have the <code>dd_bsize</code> and <code>dd_off</code> fields of SunOS 4.x. Also, SunOS release 5.5, or the ABI, SVID, or SVR4 <code>dd_loc</code> and <code>dd_size</code> fields are <code>int</code> rather than <code>long</code> as in SunOS 4.x. SunOS release 5.5, or the ABI, SVID, or SVR4 version includes <code><sys/types.h></code> while the SunOS 4.x version does not. SunOS release 5.5, or the ABI, SVID, or SVR4 version sets <code>errno</code> to <code>ENOENT</code> when the directory name argument points to an empty string.	C	C	C	N
opendir(3V) -- SysV	S		S	S	S	N
openlog(3)	S		N	N	N	N
openpl(3X)	N		N	N	N	N
optarg(3)	S		N	N	N	N
optind(3)	S		N	N	N	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
overlay(3V)	C	The SunOS 4.x <code>overlay()</code> is a function while SunOS release 5.5, or the SVID or SVR4 version of <code>overlay(srcwin, dstwin)</code> is a macro that calls <code>_overlap((srcwin), (dstwin), TRUE)</code> . The SunOS 4.x version of <code>overlay()</code> return value is undefined, while SunOS release 5.5, or the SVID or SVR4 version returns OK upon success otherwise it returns ERR.	N	C	C	S
overlay(3V) -- SysV	S		N	S	S	N
overwrite(3V)	C	The SunOS 4.x version of <code>overwrite()</code> is a function while SunOS release 5.5, or the SVID or SVR4 version of <code>overwrite(srcwin, dstwin)</code> is a macro that calls <code>_overlap((srcwin), (dstwin), FALSE)</code> . The SunOS 4.x <code>overwrite()</code> return value is undefined, while SunOS release 5.5, or the SVID or SVR4 version returns OK upon success otherwise it returns ERR.	N	C	C	S
overwrite(3V) -- SysV	S		N	S	S	N
passwd2des(3R)	S		N	N	N	N
pause(3V)	S		S	S	S	N
pclose(3S)	S		S	S	S	N
pechochar(3V) -- SysV	S		N	S	S	N
perror(3)	S		S	S	S	N
plock(3)	S		S	S	S	N
plot(3X)	S		N	N	N	N
pmap_getmaps(3N)	S	This routine is still available, but is superseded by <code>rpcb_getmaps(3N)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
<code>pmap_getport(3N)</code>	S	This routine is still available, but is superseded by <code>rpcb_getaddr(3N)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	S	N
<code>pmap_rmtcall(3N)</code>	S	This routine is still available, but is superseded by <code>rpcb_rmtcall(3N)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	S	N
<code>pmap_set(3N)</code>	S	This routine is still available, but is superseded by <code>rpcb_set(3N)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	S	N
<code>pmap_unset(3N)</code>	S	This routine is still available, but is superseded by <code>rpcb_unset(3N)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	S	N
<code>pnp(3R)</code>	N		N	N	N	N
<code>pnoutrefresh(3V) -- SysV</code>	S		N	S	S	N
<code>pod_exit(3L)</code>	N		N	N	N	N
<code>pod_getexit(3L)</code>	N		N	N	N	N
<code>pod_getmaxpri(3L)</code>	N		N	N	N	N
<code>pod_getmaxsize(3L)</code>	N		N	N	N	N
<code>pod_setexit(3L)</code>	N		N	N	N	N
<code>pod_setmaxpri(3L)</code>	N		N	N	N	N
<code>point(3X)</code>	S		N	N	N	N
<code>popen(3S)</code>	S		S	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
<code>pow(3M)</code>	C	In SunOS release 5.5, or the SVID or SVR4 version, the routine returns 0 when $x == 0$ and y is non-positive or when $x < 0$ and y not integral. For overflow or underflow, <code>pow()</code> returns +/-HUGE or 0, respectively. In both cases, <code>errno</code> is set. In the SunOS 4.x version, <code>pow(x, 0.0)</code> is 1 (which is not mentioned in SunOS release 5.5, or the SVID or SVR4); it returns NaN when $x < 0$ and y not integral, returns +/-infinity when $x == 0$ and $y < 0$. On overflow and underflow, it returns IEEE implementation-dependent values. In the SunOS release 4.x, since HUGE is defined as +Infinity, <code>pow(10.0, HUGE)</code> and <code>pow(10.0, -HUGE)</code> do not underflow or overflow and therefore no <code>errno</code> is produced. Under SunOS release 5.5, or the SVID or SVR4, these functions set <code>errno</code> to ERANGE.	N	C	C	N
<code>prefresh(3V) -- SysV</code>	S		N	S	S	N
<code>printf(3V)</code>	S		S	S	S	S
<code>printw(3V)</code>	C	SunOS release 5.5, or the SVID or SVR4 version of <code>printw()</code> returns the integer ERR upon failure and an integer value other than ERR upon successful completion. The SunOS 4.x version returns void. SunOS release 5.5, or the SVID or SVR4 headers <code><curses.h></code> automatically includes the headers <code><stdio.h></code> and <code><unctrl.h></code> and if <code>CURS_PERFORMANCE</code> is defined it defines most commonly used routines as macros for increased performance.	N	C	C	S
<code>printw(3V) -- SysV</code>	S		N	S	S	N
<code>prof(3)</code>	<code>profil()</code>	The <code>profil(2)</code> routine provides similar functionality.	A	A	A	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
psignal(3)	C	The <i>sig</i> argument is defined as an unsigned <i>int</i> in SunOS 4.x but is defined as an <i>int</i> in SVR4 and SunOS 5.5.	N	N	C	S
putc(3S)	S		S	S	S	N
putchar(3S)	S		S	S	S	N
putenv(3)	S		S	S	S	N
putp(3V) -- SysV	S		N	S	S	N
putpwent(3)	S		S	S	S	N
puts(3S)	S		S	S	S	N
putw(3S)	S		S	S	S	N
pwdauth(3)	N		N	N	N	N
qsort(3)	S		S	S	S	N
quiet_nan(3M)	N		N	N	N	N
rand(3V)	S		S	S	S	S
random(3)	drand48() and rand()	The drand48(3C) (for SunOS release 5.5, or the SVID or SVR4) or rand(3C) routines provides similar functionality.	A	A	A	S
raw(3V) -- SysV	S		N	S	S	S
rcmd(3N)	S		N	N	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
readdir(3V)	C	The <code>dirent</code> structures of SunOS 4.x <code>readdir()</code> and the ABI and SVID versions only have the <code>d_name</code> field in common. The SunOS 4.x <code>readdir()</code> supports an obsolete data structure <code>direct</code> defined in <code><sys/dir.h></code> which is no longer supported by SunOS release 5.5, or the ABI, SVID or SVR4. Applications must migrate to the <code>dirent</code> structure defined in <code><dirent.h></code> . SunOS release 5.5, or the ABI, SVID or SVR4 <code>readdir()</code> updates the directories last accessed time. The <code>dirent</code> structures of SunOS 4.x, SVR4 and SunOS 5.5 only have the <code>d_name</code> and <code>d_reclen</code> fields in common. Also, SunOS release 5.5 <code>dd_loc</code> and <code>dd_size</code> fields are type <code>int</code> rather than type <code>long</code> as in SunOS 4.x.	C	C	C	S
readdir(3V) -- SysV	C	The <code>dirent</code> structures of SunOS 4.x, SVR4 and SunOS 5.5 only have the <code>d_name</code> and <code>d_reclen</code> fields in common. Also, SunOS release 5.5 <code>dd_loc</code> and <code>dd_size</code> fields are type <code>int</code> rather than type <code>long</code> as in SunOS 4.x. SunOS release 5.5, or the ABI, SVID or SVR4 <code>readdir()</code> updates the directories last accessed time. The <code>dirent</code> structures of SunOS 4.x <code>readdir()</code> and the ABI and SVID versions only have the <code>d_name</code> field in common.	C	C	C	N
realloc(3)	C	In the SunOS release 4.x, <code>realloc()</code> accepts a pointer to a block freed since the most recent call to <code>malloc()</code> , <code>calloc()</code> , and <code>realloc()</code> . SunOS release 5.5, or the ABI, SVID, or SVR4 <code>realloc()</code> does not accept such a pointer.	C	C	C	N
realpath(3)	S		N	N	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
re_comp(3)	regexp()	For the ABI and SVID version, the <code>regexp(3)</code> general-purpose regular expression matching routines provide similar functionality. This routine is replaced by <code>recomp(3G)</code> in SunOS release 5.5.	A	A	A	S
re_exec(3)	regexp(0)	For the ABI and SVID version, the <code>regexp(3)</code> general-purpose regular expression matching routines provide similar functionality. This routine is replaced by <code>regex(3G)</code> in SunOS release 5.5.	A	A	A	S
refresh(3V) -- SysV	S		N	S	S	S
registerrpc(3N)	S	This routine is still available, but is superseded by <code>rpc_reg(3C)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	N	N	S	N
remainder(3M)	S		S	S	S	N
remexportent(3)	N	The <code>/etc/dfs/sharetab</code> file replaces <code>/etc/exports</code> in SunOS release 5.5, or the ABI, SVID, or SVR4. Refer to <code>share(1M)</code> , <code>unshare(1M)</code> , and <code>sharetab(4)</code> for more information.	N	N	N	N
remque(3)	S		N	N	S	N
resetterm(3V)	reset_shell_mode()	This routine is replaced by <code>reset_shell_mode(3)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	N	A	A	N
res_init(3)	S		N	N	S	N
res_mkquery(3)	S		N	N	S	N
res_send(3)	S		N	N	S	N
reset_prog_mode(3V) -- SysV	S		N	S	S	N
reset_shell_mode(3V) -- SysV	S		N	S	S	N
resetty(3V) -- SysV	S		N	S	S	S

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
restartterm(3V) -- SysV	S		N	S	S	N
rewind(3S)	S		S	S	S	N
rewinddir(3V)	S		S	S	S	N
rex(3R)	S		N	N	N	N
rexec(3N)	S		N	N	S	N
rindex(3)	S		A	A	A	S
rint(3M)	N		N	N	S	N
ripoffline(3V) -- SysV	S		N	S	S	N
rnusers(3R)	N		N	N	N	N
rpc_createerr(3N)	S		S	S	S	N
rpow(3X)	S		N	N	N	N
rquota(3R)	N		N	N	N	N
rresvport(3N)	S		N	N	S	N
rstat(3R)	N		N	N	N	N
rtime(3N)	N		N	N	N	S
ruserok(3N)	S		N	N	N	N
rusers(3R)	S		N	N	S	N
rwall(3R)	S		N	N	S	N
saveterm(3V)	def_prog_mode ()	This routine is replaced by def_prog_mode(3X) in SunOS release 5.5, or the ABI, SVID, or SVR4.	N	A	A	N
savetty(3V) -- SysV	S		N	S	S	S

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
scalb(3M)	C	In SunOS release 5.5, or the SVID or SVR4 version, the routine computes the value $x * (r^{**}n)$ where r is the radix of the machine's floating point arithmetic. When $r == 2$, <code>scalb()</code> is the same as <code>ldexp(3M)</code> routine. On overflow, the routine returns +/- HUGE (depending on the sign of x). On underflow, it returns 0 and sets the <code>errno</code> . In the SunOS 4.x version, the routine computes the value $x * (2^{**}n)$ at all times; <code>scalb()</code> is not defined when y is not integral.	N	C	C	N
scalbn(3M)	S		N	N	N	N
scandir(3)	N		N	N	N	S
scanf(3V)	S		S	S	S	N
scanw(3V)	C	In SunOS release 5.5, or the SVID or SVR4 the header <code><curses.h></code> automatically includes the headers <code><stdio.h></code> and <code><unctrl.h></code> and if <code>CURS_PERFORMANCE</code> is defined, it defines most commonly used routines as macros for increased performance.	N	C	C	S
scanw(3V) -- SysV	S		N	S	S	N
scr_dump(3V) -- SysV	S		N	S	S	N
scr_init(3V) -- SysV	S		N	S	S	N
scr_restore(3V) -- SysV	S		N	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
scroll(3V)	C	scroll() returns ERR on failure and an indeterminate value for success. The SunOS 4.x version returns ERR on failure and OK (0) on success. In SunOS release 5.5, or the SVID or SVR4 the header < curses.h > automatically includes the headers < stdio.h > and < unctrl.h > and if CURS_PERFORMANCE is defined it defines most commonly used routines as macros for increased performance.	N	C	C	S
scroll(3V) -- SysV	S		N	S	S	N
scrollok(3V) -- SysV	S		N	S	S	S
seconvert(3)	S		N	N	N	N
seed48(3)	S		N	S	S	N
seekdir(3V)	S		S	S	S	N
setac(3)	N		N	N	N	N
setbuf(3V)	S		S	S	S	S
setbuffer(3V)	S		N	N	N	S
set_curterm(3V) -- SysV	S		N	S	S	N
setegid(3V)	S		N	N	N	N
seteuid(3V)	S		N	N	N	N
setexportent(3)	A	The /etc/dfs/sharetab file replaces /etc/exports in SunOS release 5.5. Refer to share(1M), unshare(1M), and sharetab(4) for more information.	N	N	N	N
setfsent(3)	A	This routine is replaced by fopen(3) in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	A	N
setgid(3V)	S		S	S	S	N
setgraent(3)	N		N	N	N	N
setgrent(3V)	S		S	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
sethostent(3N)	S		N	N	S	N
setjmp(3V)	S		S	S	S	S
setkey(3)	S		N	S	S	N
setlinebuf(3V)	S		N	N	N	S
setlocale(3V)	S		S	S	S	N
setlogmask(3)	S		N	N	N	N
setmntent(3)	fopen()	The fopen(3) followed by the lockf(3) routines provide similar functionality.	A	A	A	N
setnetgrent(3N)	S		N	N	N	N
setnetent(3N)	S		N	N	S	N
setprotoent(3N)	S		N	N	S	N
setpwaent(3)	N		N	N	N	N
setpwent(3V)	S		S	S	S	N
setpwfile(3V)	N		N	N	N	N
setrgid(3V)	A	This routine is replaced by setgid(2) in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	A	N
setrpcent(3N)	S		N	N	S	N
setruid(3V)	A	This routine is replaced by setuid(2) in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	A	N
setscrreg(3V) -- SysV	S		N	S	S	N
setservent(3N)	S		N	N	S	N
setstate(3)	S		N	A	A	S
setsyx(3V) -- SysV	S		N	S	S	N
set_term(3V) -- SysV	S		N	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
setterm(3V)	C	This is an obsolete call that is replaced by <code>setupterm()</code> in both SunOS 4.x and SunOS release 5.5. See <code>curs_terminfo(3X)</code> . The call: <code>setupterm(term, 1, (int *) 0)</code> provides the same functionality as <code>setterm(term)</code> .	N	C	C	S
setterm(3V) -- SysV	S		N	S	S	N
setttyent(3)	N	Refer to <code>ttymon(1)</code> and <code>ttydefs(4)</code> for information about SunOS release 5.5 tty system.	N	N	N	N
setuid(3V)	S		S	S	S	N
setupterm(3V) -- SysV	S		N	S	S	N
setusershell(3)	S		N	N	N	N
setvbuf(3V)	S		S	S	S	S
sfconvert(3)	S		N	N	N	N
sgconvert(3)	S		N	N	N	N
sigaction(3V)	S		S	S	S	N
sigaddset(3V)	S		S	S	S	N
sigdelset(3V)	S		S	S	S	N
sigemptyset(3V)	S		S	S	S	N
sigfillset(3V)	S		S	S	S	N
sigfpe(3)	S		N	N	N	N
siginterrupt(3V)	A	The <code>sigaction(2)</code> routine provides similar functionality.	A	A	A	S
sigismember(3V)	S		S	S	S	N
siglongjmp(3V)	S		S	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
signal(3V)	C	The following SunOS 4.x signal is not defined in the SVR4 and SunOS 5.5 signal(2) routine: SIGLOST. The following SunOS 4.x signals are not defined in the ABI and SVID signal routine: SIGIO, SIGURG, SIGFSZ, SIGVTALRM, SIGPROF, SIGLOST.	C	C	C	S
ssignal(3V)	C		C	C	C	N
signaling_nan(3M)	N		N	N	N	N
signbit(3M)	N		N	N	N	N
significand(3M)	N		N	N	N	N
sigsetjmp(3V)	S		S	S	S	N
sin(3M)	C	For arguments that are much lower than zero, SunOS release 5.5, or the SVID or SVR4 version of these routines return zero because of the loss of significance. In this case, a message indicating TLOSS (see matherr(3M)) appears on the standard output. For cases of partial loss of significance, a PLOSS error is generated, but no error is printed. In both cases, errno is set to ERANGE. In the SunOS 4.x version, an argument reduction takes place for values exceeding $\pi/4$ in magnitude. The reduction could happen in software or hardware. The variable <i>fp_pi</i> defined in <math.h> allows changing of the precision at run time. The error exceptions occur in the IEEE 754 spirit for both versions.	N	C	C	N
sinh(3M)	S		N	S	S	N
single_precision(3M)	N		N	N	N	N
single_to_decimal(3)	S		N	N	N	N
sleep(3V)	S		S	S	S	S
slk_clear(3V) -- SysV	S		N	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
slk_init(3V) -- SysV	S		N	S	S	N
slk_label(3V) -- SysV	S		N	S	S	N
slk_noutrefresh(3V) -- SysV	S		N	S	S	N
slk_refresh(3V) -- SysV	S		N	S	S	N
slk_restore(3V) -- SysV	S		N	S	S	N
slk_set(3V) -- SysV	S		N	S	S	N
slk_touch(3V) -- SysV	S		N	S	S	N
sm_inter(3R)	S		N	N	N	N
space(3X)	S		N	N	N	N
spray(3R)	S		N	N	S	N
sprintf(3V)	S		S	S	S	S
sqrt(3M)	C	In the SunOS release 4.x, when <code>sqrt()</code> produces undefined results (for example, <code>sqrt(-3.0)</code>) it returns NaN, with an EDOM error and a DOMAIN matherr. SunOS release 5.5, or the SVID or SVR4 version returns 0 with an EDOM error and a DOMAIN matherr.	N	C	C	N
rand(3V)	C	In the SunOS release 4.x, argument <i>seed</i> is defined as <code>int</code> while in SunOS release 5.5, or the ABI, SVID, or SVR4 it is defined as <code>unsigned int</code> .	C	C	C	S
rand48(3)	S		N	S	S	N
random(3)	S	The <code>rand48(3C)</code> (for SunOS release 5.5, or the SVID or SVR4) or <code>rand(3C)</code> routines provide similar functionality.	A	A	A	S
sscanf(3V)	S		S	S	S	N
ssignal(3)	S		N	N	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
standard_arithmetic(3M)	N		N	N	N	N
standend(3V) -- SysV	S		N	S	S	S
standout(3V) -- SysV	S		N	S	S	S
store(3X)	dbm_store()	This routine is replaced by dbm_store(3) in SunOS release 5.5.	N	N	N	S
strcasecmp(3)	S		N	N	N	N
strcat(3)	S		S	S	S	N
strchr(3)	S		S	S	S	N
strcmp(3)	S		S	S	S	N
strcoll(3)	S		S	S	S	N
strcpy(3)	S		S	S	S	N
strcspn(3)	S		S	S	S	N
strdup(3)	S		S	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
strftime(3V)	C	<p>There are some differences in the directives specified in the format:</p> <p>%k and %l - Not supported in SunOS release 5.5.</p> <p>%S- SunOS 4.x specifies seconds to be in the range of 0-59, while SunOS release 5.5 defines seconds to be in the range of 0-61 (allows for lead seconds).</p> <p>%V, %W- Under the SunOS release 4.x, week number 01 is the first week in January with four or more days in it, while in SunOS release 5.5, week number 01 is the first week in January starting with a Sunday for %U or a Monday for %W.</p> <p>The SunOS4.1 <code>tmstructure</code> contains two fields not present in SunOS release 5.5 <code>tmstructure</code>: <code>tm_zone</code> and <code>tm_gmtoff</code>. Instead SunOS release 5.5 version uses the external variable <code>timezone</code> to contain the difference (in seconds) between GMT and local standard time, and the external variable <code>daylight</code> to indicate if daylight savings should be applied.</p> <p>Additionally, SunOS release 5.5 version uses an external variable <code>tzname</code> to store standard and summer timezone names. These external variables (<code>timezone</code>, <code>daylight</code>, and <code>tzname</code>) are supported by the SunOS 4.x System V installation option <code>ctime(3V)</code> library routines.</p>	C	C	C	N
string_to_decimal(3)	N		N	N	N	N
strlen(3)	S		S	S	S	N
strncasecmp(3)	S		N	N	N	N
strncat(3)	S		S	S	S	N
strncmp(3)	S		S	S	S	N
strncpy(3)	S		S	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
strpbrk(3)	S		S	S	S	N
strptime(3V)	A	This routine is replaced by <code>getdate(3C)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	A	N
strrchr(3)	S		S	S	S	N
strspn(3)	S		S	S	S	N
strtod(3)	C	The SunOS 4.x <code>strtod()</code> and <code>atof()</code> routines accept <code>inf_form</code> , <code>infinity_form</code> , <code>nan_form</code> , and <code>nanstring_form</code> , while SunOS release 5.5, or the ABI, SVID, or SVR4 <code>strtod()</code> and <code>atof()</code> does not accept these forms.	C	C	C	N
strtok(3)	S		S	S	S	N
strtol(3)	S		S	S	S	N
strxfrm(3)	S		S	S	S	N
stty(3C)	A	The <code>termio(7)</code> interface provides similar functionality.	A	A	A	N
subpad(3V) -- SysV	S		N	S	S	N
subwin(3V)	C	SunOS release 5.5, or the SVID or SVR4 routine returns a null pointer if failure. SunOS release 5.5, or the SVID or SVR4 header file <code><curses.h></code> automatically includes the header files <code><stdio.h></code> and <code><unctrl.h></code> and if <code>CURS_PERFORMANCE</code> is defined it defines most commonly used routines as macros for increased performance.	N	C	C	S
subwin(3V) -- SysV	S		N	S	S	N
svc_destroy(3N)	S		S	S	S	N
svc_fds(3N)	S	This routine is still available, but is superseded by <code>svc_fdset(3N)</code> in SunOS release 5.5, the SVID or SVR4.	N	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
svc_fdset(3N)	S		S	S	S	N
svc_freeargs(3N)	S		S	S	S	N
svc_getargs(3N)	S		S	S	S	N
svc_getcaller(3N)	S	This routine is still available, but is superceded by <code>svc_getrpccaller(3N)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	A	N
svc_getreq(3N)	S	This routine is still available, but is superceded by <code>svc_getreqset(3N)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	S	S	S	N
svc_getreqset(3N)	S		S	S	S	N
svc_register(3N)	<code>svc_reg()</code>	This routine is still available, but it superceded by <code>svc_reg(3N)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	A	N
svc_run(3N)	S		S	S	S	N
svc_sendreply(3N)	S		S	S	S	N
svc_unregister(3N)	<code>svc_unreg()</code>	This routine is still available, but is superceded by <code>svc_unreg(3N)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	A	N
svcerr_auth(3N)	S		S	S	S	N
svcerr_decode(3N)	S		S	S	S	N
svcerr_noproc(3N)	S		S	S	S	N
svcerr_noprogram(3N)	S		S	S	S	N
svcerr_progvers(3N)	S		S	S	S	N
svcerr_systemerr(3N)	S		S	S	S	N
svcerr_weakauth(3N)	S		S	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
svcfld_create(3N)	svc_fd_create()	This routine is still available, but is superseded by <code>svc_fd_create(3N)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	A	S
svcrow_create(3N)	S	This routine is still available, but is superseded by <code>svc_raw_create(3N)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	N	N	S	N
svctcp_create(3N)	S	This routine is still available, but is superseded by <code>svc_create(3N)</code> , <code>svc_tli_create(3N)</code> , and <code>svc_vc_create(3N)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	N	N	S	S
svcudp_bufcreate(3N)	S	This routine is still available, but is superseded by the <code>svc_tli_create(3N)</code> , and <code>svc_dg_create(3N)</code> routines in SunOS release 5.5, or the ABI, SVID, or SVR4.	N	N	S	S
svcudp_create(3N)	S	This routine is still available, but is superseded by <code>svc_create(3N)</code> , <code>svc_tli_create(3N)</code> , and <code>svc_dg_create(3N)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	N	N	S	S
swab(3)	S		S	S	S	N
sys_siglist(3)	N	Use <code>psignal(3C)</code> .	N	N	N	S
syslog(3)	S		N	N	S	N
system(3)	S		S	S	S	N
t_accept(3N)	S		S	S	S	N
t_alloc(3N)	S		S	S	S	N
t_bind(3N)	S		S	S	S	N
t_close(3N)	S		S	S	S	N
t_connect(3N)	S		S	S	S	N
t_error(3N)	S		S	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
t_free(3N)	S		S	S	S	N
t_getinfo(3N)	S		S	S	S	N
t_getstate(3N)	S		S	S	S	N
t_listen(3N)	S		S	S	S	N
t_look(3N)	S		S	S	S	N
t_open(3N)	S		S	S	S	N
t_optmgmt(3N)	S		S	S	S	N
t_rcv(3N)	S		S	S	S	N
t_rcvconnect(3N)	S		S	S	S	N
t_rcvdis(3N)	S		S	S	S	N
t_rcvrel(3N)	S		S	S	S	N
t_rcvudata(3N)	S		S	S	S	N
t_rcvuderr(3N)	S		S	S	S	N
t_snd(3N)	S		S	S	S	N
t_snddis(3N)	S		S	S	S	N
t_sndrel(3N)	S		S	S	S	N
t_sndudata(3N)	S		S	S	S	N
t_sync(3N)	S		S	S	S	N
t_unbind(3N)	S		S	S	S	N
tan(3M)	S		N	S	S	N
tanh(3M)	S		N	S	S	N
tcdrain(3V)	S		S	S	S	N
tcflow(3V)	S		S	S	S	N
tcflush(3V)	S		S	S	S	N
tcgetattr(3V)	S		S	S	S	N
tcgetpgrp(3V)	S		S	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
tcsendbreak(3V)	S		S	S	S	N
tcsetattr(3V)	S		S	S	S	N
tcsetpgrp(3V)	S		S	S	S	N
tdelete(3)	S		S	S	S	N
telldir(3V)	S		S	S	S	N
tempnam(3S)	S		S	S	S	N
textdomain(3)	N		N	N	N	N
tfind(3)	S		S	S	S	N
tgetent(3X)	C	SunOS release 5.5, or the SVID or SVR4 is supporting this routine as a conversion aid and it should not be used in new applications. SunOS release 5.5, or the SVID or SVR4 version returns ERR on failure and an integer value other than ERR upon successful completion.	N	C	C	S
tgetent(3V) -- SysV	S		N	S	S	N
tgetflag(3X)	C	SunOS release 5.5, or the SVID or SVR4 is supporting this routine as a conversion aid and it should not be used in new applications. SunOS release 5.5, or the SVID or SVR4 version returns ERR on failure and an integer value other than ERR upon successful completion.	N	C	C	S
tgetflag(3V) -- SysV	S		N	S	S	N
tgetnum(3X)	C	SunOS release 5.5, or the SVID or SVR4 is supporting this routine as a conversion aid and it should not be used in new applications. SunOS release 5.5, or the SVID or SVR4 version returns ERR on failure and an integer value other than ERR upon successful completion.	N	C	C	S
tgetnum(3V) -- SysV	S		N	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
tgetstr(3X)	C	SunOS release 5.5, or the SVID or SVR4 is supporting this routine as a conversion aid and it should not be used in new applications. SunOS release 5.5, or the SVID or SVR4 version returns ERR on failure and an integer value other than ERR upon successful completion.	N	C	C	S
tgetstr(3V) -- SysV	S		N	S	S	N
tgoto(3X)	C	SunOS release 5.5, or the SVID or SVR4 is supporting this routine as a conversion aid and it should not be used in new applications. SunOS release 5.5, or the SVID or SVR4 version returns ERR on failure and an integer value other than ERR upon successful completion.	N	C	C	S
tgoto(3V) -- SysV	S		N	S	S	N
tigetflag(3V) -- SysV	S		N	S	S	N
tigetnum(3V) -- SysV	S		N	S	S	N
tigetstr(3V) -- SysV	S		N	S	S	N
time(3V)	S		S	S	S	N
timegm(3V)	A	This routine is replaced by mktime(3C) in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	A	N
timelocal(3V)	S	This routine is replaced by localtime(3C) in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	A	N
times(3V)	C	The SunOS 4.x times() routine returns time values in units of 1/HZ seconds, where HZ is 60. SunOS release 5.5, or the ABI, SVID, or SVR4 times() routine returns time values in units of 1/CLK_TCK of a second.	C	C	C	S
timezone(3C)	S		N	N	N	N
tmpfile(3S)	C		C	C	C	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
tmpnam(3S)	S		S	S	S	N
toascii(3V)	S		S	S	S	N
toascii(3V) -- SysV	S		S	S	S	N
tolower(3V)	S		S	S	S	N
tolower(3V) -- SysV	C	SunOS release 5.5, or the ABI, SVID, or SVR4 version of this routine is affected by the program's locale as specified by LC_CTYPE, while the SunOS 4.x version is not.	C	C	C	N
touchline(3V)	C	SunOS release 5.5, or the ABI, SVID, or SVR4 version of this routine returns ERR on failure and an integer other than ERR on success.	N	C	C	S
touchline(3V) -- SysV	S		N	S	S	N
touchoverlap(3X)	N		N	N	N	S
touchwin(3V)	C	SunOS release 5.5, or the ABI, SVID, or SVR4 version of this routine returns ERR on failure and an integer other than ERR on success.	N	C	C	S
touchwin(3V) -- SysV	S		N	S	S	N
toupper(3V)	S		S	S	S	N
toupper(3V) -- SysV	C	SunOS release 5.5, or the ABI, SVID, or SVR4 version of this routine is affected by the program's locale as specified by LC_CTYPE, while the SunOS 4.x version is not.	C	C	C	N
tparam(3V) -- SysV	S		N	S	S	N
tputs(3V)	C	SunOS release 5.5, or the SVID or SVR4 is supporting this routine as a conversion aid and it should not be used in new applications. SunOS release 5.5, or the SVID or SVR4 version returns ERR on failure and an integer value other than ERR upon successful completion.	N	C	C	S
tputs(3V) -- SysV	S		N	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
traceoff(3V) -- SysV	S		N	N	S	N
traceon(3V) -- SysV	S		N	N	S	N
tsearch(3)	S		S	S	S	N
ttynname(3V)	S		S	S	S	N
ttyslot(3V)	S		N	N	S	N
twalk(3)	S		S	S	S	N
typeahead(3V) -- SysV	S		N	S	S	N
tzset(3V)	C	See <code>ctime(3V)</code> .	C	C	C	N
tzsetwall(3V)	A	This routine is replaced by <code>tzset(3C)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	A	N
ualarm(3)	S	The <code>setitimer(2)</code> system call with the <i>which</i> argument set to <code>ITIMER_REAL</code> provides similar functionality.	N	A	A	S
ulimit(3C)	S	The SVR4 and SunOS 5.5 <code>ulimit()</code> is compatible with the SunOS 4.x <code>ulimit()</code> . The SunOS 4.x version of <code>ulimit()</code> routine's integer <i>cmd</i> values 1 and 2 may not be compatible with the equivalent SVID <code>ulimit()</code> routines' symbolic constant <i>cmd</i> values <code>UL_GETFSIZE</code> and <code>UL_SETFSIZE</code> . Also, the SVID <code>ulimit()</code> routine does not support the functionality of 3 (get the maximum possible break value) and 4 (get the size of the process' file descriptor table).	C	C	S	N
unctrl(3V) -- SysV	S		N	S	S	S

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
<code>ungetc(3S)</code>	S	The SVR4 and SunOS 5.5 <code>ungetc()</code> guarantees to pushback four characters, so it is compatible with the SunOS 4.x <code>ungetc()</code> . In the SunOS release 4.x, <code>ungetc()</code> is guaranteed to pushback one character on the standard input without a previous read statement, while the ABI and SVID <code>ungetc()</code> does not support this attribute.	C	C	S	N
<code>ungetch(3V) -- SysV</code>	S		N	S	S	N
<code>user2netname(3N)</code>	S		S	S	S	N
<code>usleep(3)</code>	S	The <code>setitimer(2)</code> or <code>select(3C)</code> routines provide similar functionality.	N	A	A	S
<code>utime(3V)</code>	C	The SunOS 4.x <code>utime()</code> and SunOS release 5.5, or the ABI, SVID, or SVR4 <code>utime()</code> differ in the type of the second argument. In the SunOS release 4.x, argument <i>timep</i> points to an array of two <code>time_t</code> values, while in SunOS release 5.5, or the ABI, SVID, or SVR4 argument <i>times</i> points to <code>utimbuf</code> structure (which contains two <code>time_t</code> members).	C	C	C	N
<code>valloc(3)</code>	S		N	N	S	N
<code>varargs(3)</code>	S		N	N	N	N
<code>vfprintf(3V)</code>	C	See <code>vprintf(3V)</code> .	C	C	C	S
<code>vidattr(3V) -- SysV</code>	S		N	S	S	N
<code>vidputs(3V) -- SysV</code>	S		N	S	S	N
<code>vlimit(3C)</code>	<code>getrlimit()</code>	This routine is replaced by <code>getrlimit(2)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	A	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
vprintf(3V)	C	The SunOS 4.x vprintf(), vfprintf(), and vsprintf() routines are incompatible with SunOS release 5.5, or the ABI, SVID, or SVR4 version of these routines because of variable format list differences. In the SunOS release 4.x, <i>violist</i> (defined in <varargs.h>) is used in a function header to declare a variable argument list (for example, void function (va_alist)). Under SunOS release 5.5, or the ABI, SVID, or SVR4 the definition from <stdarg.h> is used in a function header to declare a variable argument list (for example, void function (int arg1, ...)).	C	C	C	S
vsprintf(3V)	C	See vprintf(3V).	C	C	C	S
vsyslog(3)	S	This routine is replaced by syslog(3) in SunOS release 5.5, or the ABI, SVID, or SVR4.	N	N	N	N
vtimes(3C)	getrusage()	This routine is replaced by getrusage(2) in SunOS release 5.5.	N	N	N	N
vwprintw(3V) -- SysV	S		N	S	S	N
vwscanw(3V) -- SysV	S		N	S	S	N
waddch(3V)	C	The characters in SunOS release 5.5 are chtype (long) if CHTYPE is not defined differently for < curses.h>. SunOS release 5.5, or the SVID or SVR4 version returns ERR on failure and an integer value other than ERR upon successful completion. SunOS release 5.5, or the SVID or SVR4 header file < curses.h> automatically includes the headers <stdio.h> and < unctrl.h> and if CURS_PERFORMANCE is defined it defines most commonly used routines as macros for increased performance.	N	C	C	S

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
waddch(3V) -- SysV	S		N	S	S	N
waddstr(3V)	C	SunOS release 5.5, or the SVID or SVR4 version of waddstr(3V) returns ERR (-1) on failure. SunOS release 5.5, or the SVID or SVR4 header < curses.h> automatically includes the headers <stdio.h> and <unctrl.h> and if CURS_PERFORMANCE is defined it defines most commonly used routines as macros for increased performance.	N	C	C	S
waddstr(3V) -- SysV	S		N	S	S	N
wattroff(3V) -- SysV	S		N	S	S	N
wattron(3V) -- SysV	S		N	S	S	N
wattrset(3V) -- SysV	S		N	S	S	N
wclear(3V)	C	SunOS release 5.5, or the SVID or SVR4 version of wclear() routine always returns (OK = 0) upon success while SunOS 4.x returns void. SunOS release 5.5, or the SVID or SVR4 header < curses.h> automatically includes the headers <stdio.h> and <unctrl.h> and if CURS_PERFORMANCE is defined it defines most commonly used routines as macros for increased performance.	N	C	C	S
wclear(3V) -- SysV	S		N	S	S	N
wclrtoobot(3V)	C	SunOS release 5.5, or the SVID or SVR4 version of wclrtoobot(3V) routine always returns (OK = 0) upon success while SunOS 4.x returns void. SunOS release 5.5, or the SVID or SVR4 header < curses.h> automatically includes the headers <stdio.h> and <unctrl.h> and if CURS_PERFORMANCE is defined it defines most commonly used routines as macros for increased performance.	N	C	C	S

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
wclrrobot(3V) -- SysV	S		N	S	S	N
wclrtoeol(3V)	C	SunOS release 5.5, or the SVID or SVR4 routine always returns (OK = 0) upon success while SunOS 4.x returns void. SunOS release 5.5, or the SVID or SVR4 header < curses.h > automatically includes the headers <stdio.h> and <unctrl.h> and if CURS_PERFORMANCE is defined it defines most commonly used routines as macros for increased performance.	N	C	C	S
wclrtoeol(3V) -- SysV	S		N	S	S	N
wcstombs(3)	S		S	S	S	N
wctomb(3)	S		S	S	S	N
wdelch(3V)	C	In SunOS release 5.5, or the SVID or SVR4 this routine may be a macro while it always is in SunOS 4.x. SunOS release 5.5, or the SVID or SVR4 version returns ERR on failure and an integer value other than ERR upon successful completion.	N	C	C	S
wdelch(3V) -- SysV	S		N	S	S	N
wdeleteln(3V)	C	In SunOS release 5.5, or the SVID or SVR4 this routine may be a macro while it always is in SunOS 4.x. SunOS release 5.5, or the SVID or SVR4 version returns ERR on failure and an integer value other than ERR upon successful completion.	N	C	C	S
wdeleteln(3V) -- SysV	S		N	S	S	N
wechochar(3V) -- SysV	S		N	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
werase(3V)	C	In SunOS release 5.5, or the SVID or SVR4 this routine returns OK (0) or a non-negative integer if <code>immedok</code> is set. SunOS release 5.5, or the SVID or SVR4 header <code><curses.h></code> automatically includes the headers <code><stdio.h></code> and <code><unctrl.h></code> and if <code>CURS_PERFORMANCE</code> is defined it defines most commonly used routines as macros for increased performance.	N	C	C	S
werase(3V) -- SysV	S		N	S	S	N
wgetch(3V)	C	SunOS release 5.5, or the SVID or SVR4 version of <code>wgetch()</code> returns ERR on failure and an integer value other than ERR upon successful completion. SunOS release 5.5, or the SVID or SVR4 version also has additional support for function keys.	N	C	C	S
wgetch(3V) -- SysV	C	In SunOS release 5.5, or the SVID or SVR4, if the window is not a pad, and it has been moved or modified since the last call to <code>wrefresh()</code> , <code>wrefresh()</code> will be called before another character is read. In the SunOS release 4.x, <code>wrefresh()</code> will not be called under these circumstances.	N	C	C	N
wgetstr(3V)	C	SunOS release 5.5, or the SVID or SVR4 version of <code>wgetstr()</code> returns ERR on failure and an integer value other than ERR upon successful completion.	N	C	C	S
wgetstr(3V) -- SysV	C	See <code>getstr(3V)</code> — Sys V.	N	C	C	N
winch(3V) -- SysV	S		N	S	S	S
winsch(3V)	C	SunOS release 5.5, or the SVID or SVR4 version of <code>winsch()</code> returns ERR on failure and an integer value other than ERR upon successful completion.	N	C	C	S
winsch(3V) -- SysV	S		N	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
winsertln(3V)	C	SunOS release 5.5, or the SVID or SVR4 version of <code>winsertln()</code> returns ERR on failure and an integer value other than ERR upon successful completion. This can be a macro in SunOS release 5.5, or the SVID or SVR4.	N	C	C	S
winsertln(3V) -- SysV	S		N	S	S	N
wmove(3V)	C	SunOS release 5.5, or the SVID or SVR4 version of <code>wmove()</code> returns ERR on failure and an integer value other than ERR upon successful completion.	N	C	C	S
wmove(3V) -- SysV	S		N	S	S	N
wnoutrefresh(3V) -- SysV	S		N	S	S	N
wprintw(3V)	C	SunOS release 5.5, or the SVID or SVR4 version of <code>wprintw()</code> returns ERR on failure and an integer value other than ERR upon successful completion. The SunOS 4.x version returns <code>void</code> . SunOS release 5.5, or the SVID or SVR4 header <code><curses.h></code> automatically includes the headers <code><stdio.h></code> and <code><unctrl.h></code> and if <code>CURS_PERFORMANCE</code> is defined it defines most commonly used routines as macros for increased performance.	N	C	C	S
wprintw(3V) -- SysV	S		N	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
wrefresh(3V)	C	SunOS release 5.5, or the SVID or SVR4 version of <code>wrefresh()</code> returns (ERR = -1) on failure and some other integer on success while SunOS 4.x returns <code>void</code> . SunOS release 5.5, or the SVID or SVR4 header <code><curses.h></code> automatically includes the headers <code><stdio.h></code> and <code><unctrl.h></code> and if <code>CURS_PERFORMANCE</code> is defined it defines most commonly used routines as macros for increased performance.	N	C	C	S
wrefresh(3V) -- SysV	S		N	S	S	N
wscanw(3V)	C	SunOS release 5.5, or the SVID or SVR4 version of <code>wscanw()</code> returns an <code>int</code> containing the number of fields mapped by the call while SunOS 4.x returns <code>void</code> . SunOS release 5.5, or the SVID or SVR4 header <code><curses.h></code> automatically includes the headers <code><stdio.h></code> and <code><unctrl.h></code> and if <code>CURS_PERFORMANCE</code> is defined it defines most commonly used routines as macros for increased performance.	N	C	C	S
wscanw(3V) -- SysV	S		N	S	S	N
wsetscrreg(3V) -- SysV	S		N	S	S	N
wstandend(3V)	C	This is a <code>curses(3V)</code> function which clears all window attributes using <code>attrset(0)</code> . The SunOS 4.x version always returns undefined while SunOS release 5.5, or the SVID or SVR4 <code>standout()</code> routine always returns 1 (success).	N	C	C	S
wstandend(3V) -- SysV	S		N	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
wstandout(3V)	C	This is a <code>curses(3V)</code> function which sets the <code>A_STANDOUT</code> attribute to enable the terminals best standout mode for a window. The SunOS 4.x version uses <code>attron(A_STANDOUT)</code> for this function and returns undefined. SunOS release 5.5, or the SVID or SVR4 <code>standout()</code> routine is the same as: <code>attron(A_STANDOUT)</code> and always returns 1 (success).	N	C	C	S
wstandout(3V) -- SysV	S		N	S	S	N
xcrypt(3R)	N		N	N	N	N
xdecrypt(3R)	N		N	N	N	N
xdr_accepted_reply(3N)	S		S	S	S	N
xdr_array(3N)	S		S	S	S	N
xdr_authunix_parms(3N)	S	This routine is still available, but is superseded by <code>xdr_authsys_parms(3N)</code> in SunOS release 5.5, or the ABI, SVID, or SVR4.	A	A	A	N
xdr_bool(3N)	S		S	S	S	N
xdr_bytes(3N)	S		S	S	S	N
xdr_callhdr(3N)	S		S	S	S	N
xdr_callmsg(3N)	S		S	S	S	N
xdr_char(3N)	S		S	S	S	N
xdr_destroy(3N)	S		S	S	S	N
xdr_double(3N)	S		S	S	S	N
xdr_enum(3N)	S		S	S	S	N
xdr_float(3N)	S		S	S	S	N
xdr_free(3N)	S		S	S	S	N
xdr_getpos(3N)	S		S	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
xdr_inline(3N)	S		S	S	S	N
xdr_int(3N)	S		S	S	S	N
xdr_long(3N)	S		S	S	S	N
xdr_opaque(3N)	S		S	S	S	N
xdr_pointer(3N)	S		S	S	S	N
xdr_reference(3N)	S		S	S	S	N
xdr_setpos(3N)	S		S	S	S	N
xdr_short(3N)	S		S	S	S	N
xdr_string(3N)	S		S	S	S	N
xdr_u_char(3N)	S		S	S	S	N
xdr_u_int(3N)	S		S	N	S	N
xdr_u_long(3N)	S		S	S	S	N
xdr_u_short(3N)	S		S	S	S	N
xdr_union(3N)	S		S	S	S	N
xdr_vector(3N)	S		S	S	S	N
xdr_void(3N)	S		S	S	S	N
xdr_wrapstring(3N)	S		S	S	S	N
xdrmem_create(3N)	S		S	S	S	N
xdrrec_create(3N)	S		S	S	S	N
xdrrec_endofrecord(3N)	S		S	N	S	N
xdrrec_eof(3N)	S		S	S	S	N
xdrrec_skiprecord(3N)	S		S	N	S	N
xdrstdio_create(3N)	S		S	S	S	N
xtom(3X)	S		N	N	N	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes	ABI	SVID	SVR4	BSD
y0(3M)	C	In the SunOS release 4.x, when these routines have undefined results they return NaN, with an EDOM error and a DOMAIN matherr. In SunOS release 5.5, or the SVID or SVR4, they return -HUGE with an EDOM error and a DOMAIN matherr. In the SunOS release 4.x, <code>y0(HUGE)</code> , <code>y1(HUGE)</code> , <code>yn(9, HUGE)</code> will return zero with no error indication.	N	C	C	N
y1(3M)	C	In the SunOS release 4.x, when these routines have undefined results they return NaN, with an EDOM error and a DOMAIN matherr. In SunOS release 5.5, or the SVID or SVR4, they return -HUGE with an EDOM error and a DOMAIN matherr. In the SunOS release 4.x, <code>y0(HUGE)</code> , <code>y1(HUGE)</code> , <code>yn(9, HUGE)</code> will return zero with no error indication.	N	C	C	N
yn(3M)	C	In the SunOS release 4.x, when these routines have undefined results they return NaN, with an EDOM error and a DOMAIN matherr. In SunOS release 5.5, or the SVID or SVR4, they return -HUGE with an EDOM error and a DOMAIN matherr. In the SunOS release 4.x, <code>y0(HUGE)</code> , <code>y1(HUGE)</code> , <code>yn(9, HUGE)</code> will return zero with no error indication.	N	C	C	N

System Files Reference Table



This appendix contains the System Files reference table. This table lists all SunOS 4.x system files, and shows their status in the Solaris 2.5 environment.

Using the Reference Table

- If an interface is listed as “changed” (C), a brief description of differences between the SunOS 4.x command and the Solaris 2.5 command is provided.
- If an interface is listed as “the same” (S), the Solaris 2.5 interface supports all features of the SunOS 4.x interface. In some cases the interface has been enhanced, but can be considered a complete superset of the SunOS 4.x interface.
- If an interface is listed as “not available” (N), check the Notes section for information about its replacement.

For complete information on all Solaris 2.5 interfaces, see the *man Pages(4): File Formats*.

≡ D

System Files

Table D-1 File Formats Reference Table

SunOS release 4.x	SunOS release 5.5	Notes
a.out(5)	C	Assembler and link editor output format
acct(5)	S	Execution accounting file
aliases(5)	S	Addresses and aliases for sendmail
ar(5)	S	Archive (library) file format
audit.log(5)	N	The security audit trail file
audit_control(5)	N	Control information for system audit daemon
audit_data(5)	N	Current information on audit daemon
auto.home(5)	C	Automount map for home directories
auto.master(5)	C	Automount map for home directories
bar(5)	N	Tape archive file format
boards.pc(5)	N	ATN and XTN compatible boards for DOS windows
bootparams(5)	S	Boot parameter data base
cpio(5)	S	Format of cpio archive
crontab(5)	S	Table of times to run periodic jobs
dir(5)	A	Format of directories
dump(5)	C	Incremental dump format
environ(5V)	C	User environment
ethers(5)	S	Ethernet address to <i>hostname</i> database or NIS domain
exports(5)	A	Directories to export to NFS clients
fbtab(5)	C	Framebuffer table
fcntl(5)	C	File control options
fs(5)	C	Format of a 4.2 (ufs) file system volume

Table D-1 File Formats Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes
<code>fspec(5)</code>	S	Format specification in text files
<code>fstab(5)</code>	A	Static filesystem mounting table, mounted filesystems table
<code>ftpusers(5)</code>	S	List of users prohibited by FTP
<code>gettytab(5)</code>	N	Terminal configuration data base
<code>group(5)</code>	S	Group file
<code>group.adjunct(5)</code>	N	Group security data file
<code>holidays(5)</code>	C	Prime/non-prime table for System V accounting
<code>hosts(5)</code>	S	Hostname data base
<code>hosts.equiv(5)</code>	S	Trusted hosts by system and by user
<code>indent.pro(5)</code>	N	Default options for indent
<code>inetd.conf(5)</code>	S	Internet servers database
<code>internat(5)</code>	N	Key mapping table for internationalization
<code>keytables(5)</code>	S	Keyboard table descriptions for loadkeys and dumpkeys
<code>link(5)</code>	N	Link editor interfaces
<code>locale(5)</code>		Locale database
<code>magic(5)</code>	S	File command's magic number file
<code>mtab(5)</code>	A	Mounted file system table
<code>netgroup(5)</code>	S	List of network groups
<code>netmasks(5)</code>	S	Network mask data base
<code>netrc(5)</code>	S	File for ftp remote login data
<code>networks(5)</code>	S	Network name data base
<code>passwd(5)</code>	C	Password file
<code>passwd.adjunct(5)</code>	N	User security data file. See <code>shadow(4)</code> .

≡ D

Table D-1 File Formats Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes
phones(5)	S	Remote host phone number data base
plot(5)	N	Graphics interface
printcap(5)	A	Printer capability data base
proto(5)	S	Prototype job file for at
protocols(5)	S	Protocol name data base
publickey(5)	S	Public key database
queuedefs(5)	S	Queue description file for at, batch, and cron
rasterfile(5)	S	Sun's file format for raster images
remote(5)	S	Remote host description file
resolv.conf(5)	S	Configuration file for domain name system resolver
rfmaster(5)	N	
rgb(5)	N	Available colors (by name) for colordit
rhosts(5)	S	Trusted hosts by system and by user
rmtab(5)	S	Remote mounted file system table
rootmenu(5)	A	Root menu specification for SunView
rpc(5)	S	RPC program number data base
sccsfile(5)	S	Format of an SCCS history file
services(5)	S	Internet services and aliases
sm(5)	S	in.statd directory and file structures
statmon(5)	S	statd directories and file structures
sunview(5)	A	Initialization file for SunView
svdtab(5)	N	SunView device table
syslog.conf(5)	S	Configuration file for syslogd system log daemon
systems(5)	C	NIS systems file

Table D-1 File Formats Reference Table (Continued)

SunOS release 4.x	SunOS release 5.5	Notes
tar(5)	S	Tape archive file format
term(5)	S	Terminal driving tables for nroff
term(5V)	S	Format of compiled term file
termcap(5)	S	Terminal capability data base
terminfo(5V)	S	Terminal capability data base
toc(5)	N	Table of contents of optional clusters
translate(5)	N	Input and output files for system message translation
ttytab(5)	N	Terminal initialization data
types(5)	S	Primitive system data types
tzfile(5)	S	Time zone information
updaters(5)	S	Configuration file for NIS updating
utmp(5V)	C	Login records
uuencode(5)	S	Format of an encoded uuencode file
vfont(5)	N	Font formats
vgrind(5)	N	vgrind's language definition data base
xtab(5)	N	Directories to export to NFS clients
ypaliases(5)	N	NIS aliases for sendmail
ypfiles(5)	S	NIS database and directory structure
ypgroup(5)	N	NIS group file
yppasswd(5)	N	NIS password file
ypprintcap(5)	N	NIS printer capability database

/and/usr File Systems Changes



This Appendix shows the layout of directories in the / and /usr file systems. Appendix A, “Commands Reference Table”, explains differences in commands within these directories.

Layout of the / File System

Table E-1 shows the layout of the SunOS release 5.5 / file system, which contains directories that are unique to each system.

Table E-1 Directories in the / File System

Directory	Description
/	Root of the overall file system name space
/dev	Primary location for special files
/dev/dsk	Block disk devices
/dev/rdsk	Raw disk devices
/dev/pts	Pseudo terminal slave devices
/dev/rmt	Raw tape devices
/dev/sad	Entry points for the STREAMS Administrative Driver
/dev/term	Terminal devices
/etc	Host-specific system administrative configuration files and databases
/etc/acct	Accounting system configuration information
/etc/cron.d	Configuration information and FIFO for cron

Table E-1 Directories in the / File System (Continued)

Directory	Description
/etc/default	Default information for various programs
/etc/dfs	Configuration information for exported file systems
/etc/fs	Binaries organized by file system types for operations required before /usr is mounted
/etc/inet	Configuration files for Internet services
/etc/init.d	Scripts for transitioning among run levels
/etc/lib	Shared libraries needed during booting
/etc/lp	Configuration information for the printer subsystem
/etc/mail	Mail subsystem configuration
/etc/net	Configuration information for ti (transport independent) network services
/etc/opt	Configuration information for optional packages
/etc/rc0.d	Scripts for entering or leaving run level 0
/etc/rc1.d	Scripts for entering or leaving run level 1
/etc/rc2.d	Scripts for entering or leaving run level 2
/etc/rc3.d	Scripts for entering or leaving run level 3
/etc/rcS.d	Scripts for entering or leaving run level S
/etc/saf	Service Access Facility (SAF) files, including FIFOs
/etc/skel	Default profile scripts for new user accounts
/etc/sm	Status monitor information
/etc/sm.bak	Backup copy of status monitor information
/etc/tm	Trademark files; contents displayed at boot time
/etc/uucp	Configuration information for uucp
/export	Default root of the exported file system tree
/home	Default root of a subtree for user directories
/kernel	Subtree of loadable kernel modules, including the base kernel itself as /kernel/unix
/mnt	Temporary mount point for file systems
/opt	Root of a subtree for add-on application packages
/opt/SUNWspro	Mount/installation point for unbundled language products
/platform	Subtree of loadable kernel modules

Table E-1 Directories in the / File System (Continued)

Directory	Description
/sbin	Essential executables used in the booting process and in manual system failure recovery
/tmp	Temporary files; cleared during boot sequence
/usr	Mount point for the /usr file system
/var	Root of a subtree of various files
/var/adm	System logging and accounting files
/var/crash	Default depository for kernel crash dumps
/var/cron	Log file for cron
/var/lp	Line printer subsystem logging information
/var/mail	Directory where users' mail is kept
/var/news	Community service messages (not to be confused with USENET-style news)
/var/nis	NIS+ databases
/var/opt	Root of a subtree for various files associated with optional software packages
/var/options	Provides package compatibility with pre-SunOS 5.0 packages
/var/preserve	Backup files for vi and ex editors
/var/sadm	Databases maintained by the software package management utilities
/var/saf	System Access Facility (SAF) logging and accounting files
/var/spool	Directories for spooled temporary files
/var/spool/cron	Spool files for cron and at
/var/spool/locks	Spooling lock files
/var/spool/lp	Line printer spool files
/var/spool/mqueue	Mail queued for delivery
/var/spool/pkg	Spoiled packages
/var/spool/uucp	Queued uucp jobs
/var/spool/uucppublic	Files deposited by uucp

Table E-1 Directories in the / File System (Continued)

Directory	Description
/var/tmp	Directory for temporary files not cleared during boot sequence
/var/uucp	Log and status files for uucp
/var/yp	Databases for yp (for backward compatibility with NIS and ypbind)

Layout of the /usr File System

Table E-2 shows the layout of the /usr file system, which contains architecture-dependent and architecture-independent sharable files.

Table E-2 Directories in the /usr File System

Directory	Description
/usr/4lib	Libraries for the binary compatibility a.out package (BCP)
/usr/bin	Location for standard system commands
/usr/bin/sunview1	SunView executables, part of BCP
/usr/ccs	The C compilation system
/usr/ccs/bin	Binaries
/usr/ccs/lib	Libraries and auxiliary files
/usr/demo	Demo programs and data
/usr/games	Game binaries and data
/usr/include	Include header files (for C programs, etc.)
/usr/kernel	additional modules
/usr/kvm	Implementation architecture-specific binaries and libraries
/usr/lib	Various program libraries, architecture-dependent databases, and binaries not invoked directly by the user
/usr/lib/acct	Accounting scripts and binaries
/usr/lib/dict	Database files for the spell command
/usr/lib/class	Scheduling class-specific directories containing executables for priocntl and dispadmin commands

Table E-2 Directories in the /usr File System (Continued)

Directory	Description
/usr/lib/font	Font description files for <code>troff</code>
/usr/lib/fs	File system type dependent modules; not invoked directly by the user
/usr/lib/iconv	Conversion tables for <code>iconv</code>
/usr/lib/libp	Profiled libraries
/usr/lib/locale	Internationalization and localization databases
/usr/lib/lp	Line printer subsystem databases and backend executables
/usr/lib/mail	Auxiliary programs for the mail subsystem
/usr/lib/netsvc	Internet network services
/usr/lib/nfs	Auxiliary NFS [®] -related programs and daemons
/usr/lib/pics	PIC archives needed to build the runtime linker
/usr/lib/refer	Preprocessor for <code>nroff/troff</code>
/usr/lib/sa	Scripts and commands for the system activity report package
/usr/lib/saf	Auxiliary programs and daemons related to the Service Access Facility (SAF)
/usr/lib/spell	Auxiliary <code>spell</code> -related programs and databases
/usr/lib/uucp	Auxiliary <code>uucp</code> -related programs and daemons
/usr/local	Commands local to a site
/usr/net/servers	Entry points for foreign name service requests related by the listener
/usr/oasys	Files pertaining to the optional FACE package
/usr/old	Programs that are being phased out
/usr/openwin	Mount or installation point for OpenWindows software
/usr/sadm	Various files and directories related to system administration
/usr/sadm/bin	Binaries for use by FMLI scripts
/usr/sadm/install	Executables and scripts for package management
/usr/sbin	Executables for system administration

Table E-2 Directories in the /usr File System (Continued)

Directory	Description
/usr/sbin/static	Statically linked versions of selected programs from /usr/bin and /usr/sbin; used to recover from broken dynamic linking
/usr/share	Architecture-independent databases
/usr/share/lib	Architecture-independent databases
/usr/share/lib/keytables	Keyboard layout description tables
/usr/share/lib/mailx	Help files for mailx
/usr/share/lib/nterm	Terminal tables for nroff
/usr/share/lib/pub	Various data files
/usr/share/lib/spell	Auxiliary spell-related databases and scripts
/usr/share/lib/tabset	Tab setting escape sequences
/usr/share/lib/terminfo	Terminal description files
/usr/share/lib/tmac	Macro packages for nroff and troff
/usr/share/lib/zoneinfo	Time zone information
/usr/share/src	Source code for kernel, libraries, and utilities
/usr/snadm	Files associated with Administration Tool (admintool)
/usr/ucb	Berkeley compatibility package binaries
/usr/ucbinclude	Berkeley compatibility package header files
/usr/ucblib	Berkeley compatibility package libraries
/usr/vmsys	Files pertaining to the optional FACE package

Quick Reference for Basic Changes



This appendix is a quick reference for changes in common commands, files and directories, and daemons and standard processes.

Table F-1 Basic Commands

Basic Commands		
SunOS Release 4.x	Solaris 2.5	Comments
lpr	lp	Basic default print command
lpr -P<printer>	lp -d<printer>	Specifying a printer with the print command
lpq	lpstat -o	Check the print queue of the default printer
lpq -P<printer>	lpstat -o<printer>	Check the status of a specific printer and list print IDs
	lpstat -a	Determine which printers are available (in SunOS release 4.x, you would check the /etc/printcap file)
lprm <print job#>	cancel <request ID>	Cancel a print job
	cancel <printer>	Alternate method for canceling a currently active print job
ps -ax	ps -ef	Process status is the same, but some of the options have changed
pstat -s	swap -s	Prints information about swap space

Table F-2 Advanced Commands

Advanced Commands		
SunOS Release 4.x	Solaris 2.5	Comments
exportfs	share <resource>	Used to make specified resource listed in user's /etc/dfs/dfstab available for remote mount
exportfs -a	shareall	Option to make all resources listed in user's /etc/dfs/dfstab available for mounting
exportfs -u	unshare <resource>	Used to make resources unavailable
mount -a	mountall	Mount all file systems specified in /etc/vfstab, where the mountall option is set
umount -a	umountall	Unmount all file systems in /etc/vfstab, other than root, /proc, /var, and /usr
showmount -d	dfmounts <option>	Lists mounted NFS file systems where option specifies machine name
showmount -e	dfshares <option>	Lists shared (exported) NFS file systems
dump	ufsdump	For backing up file systems or specified files
restore	ufsrestore	For restoring files dumped to backup media

Table F-3 Files and Directories

Files and Directories		
SunOS Release 4.x	Solaris 2.5	Comments
/var/spool/mail	/var/mail	Location for incoming mail
/etc/fstab	/etc/vfstab	File system mount table
/etc/exports	/etc/dfs/dfstab	Lists exported file systems
/etc/mtab	/etc/mnttab	List of currently mounted resources read by the /etc/mount command
/etc/xtab	/etc/dfs/sharetab	List of shareable resources
/usr/bin	/usr/bin and /usr/sbin	/usr/sbin is available with Solaris executables
/etc/aliases	/etc/mail/aliases	New location for local E-mail alias file
/etc/printcap	No longer exists	Capability replaced by /usr/share/lib/terminfo and files in /etc/lp
/etc/passwd	/etc/passwd /etc/shadow	Capability is shared with counterpart, the /etc/shadow file, which stores user's encrypted passwords and other information

Table F-4 Daemons and Standard Processes

Daemons and Standard Processes		
SunOS Release 4.x	Solaris 2.5	Comments
/usr/lib/lpd	/usr/lib/lp/lpsched	Print daemon
/usr/etc/rpc.lockd	/usr/lib/nfs/lockd	Network lock daemon
/usr/etc/rpc.mountd	/usr/lib/nfs/mountd	NFS mount request server
/usr/etc/ypbind	/usr/lib/netsvc/yp/ypbind	NIS binder process
/usr/etc/nfsd	/usr/lib/nfs/nfsd	NFS daemon
/usr/etc/biod	No longer exists	Block I/O daemon Capability implemented in the kernel
/etc/rc and /etc/rc.local	/etc/rc[012356S].d	System initialization scripts

Glossary

Architecture

The specific components of a computer system and the way they interact with one another. From a Solaris 2.5 kernel perspective, “architecture” refers to the type of CPU chip in the system. In this manual, the only architecture discussed is the kernel architecture (for example `sun4`, `sun4c`, or `sun4m`).

Binary Compatibility Package

An optional package that allows existing SunOS release 4.x applications, both statically and dynamically linked, to run under SunOS release 5.5 without modification or recompilation.

Client

A system that uses NIS, NFS, or other services provided by another system.

Cluster

A functional collection of software packages.

Configuration cluster

A default selection of clusters representing typical software selections.

Dataless

A system whose `/usr` and `/usr/kvm` file systems are provided by a file server, and whose root and swap disk partitions are on a directly connected disk.

DDI

Device Driver Interface. Facilitates both source and binary portability across successive releases of the operating system on a particular system.

DKI

Driver Kernel Interface. A defined service interface for the entry point routines and utility functions specified for communication between the driver and the kernel. It does not encompass the driver/hardware or the driver/boot software interface.

Disk partition

See disk slice.

Disk slice

A discrete portion of a disk, configured during installation. Slices were referred to as partitions under SunOS 4.1.x and System V Release 3.

Diskless

A system whose `root`, `swap`, and `/usr` file systems (disk partitions) are provided by an NFS server (or file server) instead of a directly connected disk.

DNS

Domain name system. The distributed name/address mechanism used in the Internet.

ELF

Executable and linking format. The native object format of Solaris 2.5 executables.

Heterogeneous server

A server of diskless clients that is a mix of its own architecture and other kernel architectures.

Homogeneous server

A server of diskless clients that has only clients with the same kernel architecture.

Install server

A machine that provides boot service and network access to the Solaris 2.5 distribution. This can be either on a local CD-ROM or a file system containing a copy of the distribution.

IP address

A unique number that identifies each host in a network. The address is partitioned into two distinct parts: a network part and a host part.

Kernel architecture

The hardware portion of a Solaris 2.5 kernel. Two systems have the same kernel architecture if the same Solaris 2.5 kernel runs on both of them. Not all Sun-4 systems have the same kernel architecture.

Multiple OS operation

The operation that enables a SPARC server to continue serving SunOS 4.1.x clients while the server is running the Solaris 2.3 software release. In this special case, a heterogeneous server could be serving clients of the same kernel architecture.

Netmask

A number used by software to separate additional network information (called the “subnet”) from the host part of an IP address. The netmask is also referred to as the subnet mask.

NIS

The network information service. NIS provides information about machines and services in a local area network.

NIS+

An enhanced version of the network information service software. These enhancements include secure updates, better performance, and hierarchical naming.

OLIT

Abbreviation for OPEN LOOK Intrinsic Toolkit.

Package

A functional grouping of software. All SunOS release 5.5 software is grouped and distributed in packages. Packages are also the standard way to deliver unbundled Sun and third-party software.

SAC

Services Access Facility. A SunOS release 5.5 tool for managing access to local and network system services, such as modems and terminals

SAF

Service Access Control. Commands used to set up and manage services.

Server

A system that provides services to the network. These services include NFS system and NIS database access.

Source Compatibility Package

An optional package that contains a collection of SunOS release 4.x and BSD commands, library routines, and header files otherwise not available with Solaris 2.5 software.

Standalone

A system that does not depend on a server for its `root`, `swap`, or `/usr` disk partitions.

Time zone

Any of the 24 longitudinal divisions of the earth's surface for which a standard time is kept.

Unbundled

Software products not delivered as part of SunOS release 5.5 software distribution: for example, the SunPro compilers.

Index

Symbols

#, xxv
\$, xxv
%, xxv
/ file system, 28, 81, 339 to 342

Numerics

/4lib directory, 342
/5bin directory, 93
/5include directory, 93
/5lib directory, 93, 162, 163

A

a.out system file, 94, 334
a641 library routine, 266
abort library routine, 266
abs library routine, 266
ac command, 198
accept system call, 239
access system call, 239
accounting files
 scripts and binaries, 342
 Service Access Facility (SAF), 341
 system, 341
 UUCP, 129

accounting system
 See also system accounting; user
 accounts
 configuration information
 directory, 339
acct command, 226
/acct directory, 339, 342
acct system call, 239
acct system file, 334
acctcms command, 198
acctcom command, 198
acctcon1 command, 198
acctcon2 command, 198
acctdisk command, 198
acctdusg command, 198, 207
acctmerg command, 198
accton command, 198
acctprc1 command, 198
acctprc2 command, 198
acctwtmp command, 198
acos library routine, 266
acosh library routine, 266
adb command, 154, 155, 198, 199
adbgen command, 198
add_client command, 198
add_drv command, 72, 180, 181, 193

add_services command, 198
 addbib command, 198
 addch library routine, 266
 addexportent library routine, 266
 adding
 devices, 65 to 66, 181, 193
 groups, 53
 network devices, 21 to 22
 NIS+ information, 20
 port monitor services, 123
 port monitors, 123
 software packages, 12, 167
 users, 53
 addmntent library routine, 266
 addstr library routine, 266
 adjacentscreens command, 198
 adjtime system call, 239
 /adm directory, 341
 admin command, 168, 199
 /.Admin directory, 129
 administrators, *See* system administrators
 Administration Tool
 See also admintool command; Host
 Manager; User Account
 Manager
 described, 8
 files directory, 93, 344
 Serial Port Manager, 121 to 122
 Admintool
 described, 18 to ??
 admintool command, 18 to ??
 See also Administration Tool
 add_client command vs., 198
 files directory, 344
 rm_client command vs., 224
 adv command, 199
 aedplot command, 199
 aging passwords, 44, 45
 agt_create library routine, 266
 agt_enumerate library routine, 266
 agt_trap library routine, 266
 aint library routine, 266
 aiocancel library routine, 266
 aioread library routine, 266
 aiowait library routine, 266
 aiowrite library routine, 266
 alarm library routine, 266
 aliases system file, 334, 347
 align_equals command, 199
 alloca library routine, 266
 alphasort library routine, 267
 analyze command, 199
 anint library routine, 267
 annuity library routine, 267
 ANSI C compiler, 139 to 146
 application concurrency, 8, 190
 applications
 determining compatibility of, 165 to
 166
 porting, reasons for, 40
 apropos command, 199
 ar command, 168, 199, 223
 ar system file, 334
 arc library routine, 267
 arch command, 199
 architecture-independent databases,
 directories for, 344
 architecture-specific information,
 directories for, 342
 archives, converting to random
 libraries, 169
 arp command, 199
 as command, 168, 199
 asctime library routine, 267
 ASET (automated security enhancement
 tool), 8, 46 to 47
 Asian language character sets, 173, 174
 asin library routine, 267
 asinh library routine, 267
 assert library routine, 267
 async_daemon system call, 239
 at command, 199, 214, 341
 AT&T systems

device naming, 14
 SVR4 features excluded from Solaris
 operating environment, 9 to
 10
 sysadm menu utility, 9 to 10
 atan library routine, 267
 atan2 library routine, 267
 atanh library routine, 267
 atexit library routine, 299
 atof library routine, 267
 atoi library routine, 267
 atol library routine, 267
 atoplot command, 199
 atq command, 200
 atrm command, 200
 attroff library routine, 267
 attron library routine, 267
 attrset library routine, 268
Audio Tool, 4
 audioio ioctl's, 159
 audit command, 200
 audit system call, 240
 audit.log system file, 334
 audit_args library routine, 268
 audit_control system file, 334
 audit_data system file, 334
 audit_text library routine, 268
 audit_warn command, 200
 auditd command, 200
 auditon system call, 240
 auditsvc system call, 240
 auth_destroy library routine, 268
 authdes_create library routine, 268
 authdes_getucred library routine, 268
 authdes_seccreate library
 routine, 268
 authnone_create library routine, 268
 authsys_create_default library
 routine, 268
 authsys_seccreate library
 routine, 268
 authunix_create library routine, 268
 authunix_create_default library
 routine, 268
 auto configuration, 8
 auto.home system file, 200, 334
 auto.master system file, 200, 334
 auto_home system file, 200
 auto_master system file, 200
 autofs program, 17 to 18, 95 to 96
 automated security enhancement tool
 (ASET), 8, 46 to 47
 automatic loading, device drivers, 9
 automount command, 200
 automounting, 17 to 18, 95 to 96, 97, 200
 autopush command, 192
 awk command, 200

B

backing up
 See also cpio command; dd command;
 dump command; restoring;
 tar command; ufsdump
 command
 changes in, 101, 102 to 105
 installation and, 25 to 26, 30
Backup CoPilot, 26, 101
backup files, directory for vi and ex
 editors, 341
 banner command, 200
 bar command, 101, 200
 bar system file, 334
 basename command, 201
 batch command, 201
 baudrate library routine, 268
 bc command, 201
 bcmp library routine, 268
 bcopy library routine, 268
 beep library routine, 268
Berkeley compatibility package,
 directories for, 344
BFS (boot file system), 84

bgplot command, 201
 biff command, 201
 /bin directory, *See* /usr/bin directory
 binaries, directories for, 340, 342, 343
 Binary Compatibility Package, 39, 40 to 41, 165 to 166, 342
 bind system call, 240
 bindresvport library routine, 268
 bindtextdomain library routine, 175
 bin-mail command, 201
 biod command, 201, 347
 block disk devices, directory for, 87, 183, 339
 block I/O daemon, 347
 boards.pc system file, 334
 boot block, 70, 179 to 180
 boot command
 changes in, 70, 71, 179 to 180, 201
 device additions and, 65 to 66, 70
 reconfiguration boot, 181
 boot file system (BFS), 84
 boot.sun4c.sunos.4.1 command, 71, 180
 bootblk command, 180
 booting, 69 to 72, 179 to 180
 See also boot command
 changes in, 69 to 70, 71 to 72, 179 to 180
 from PROM, 70 to 71
 bootparam library routine, 268
 bootparamd command, 201
 bootparams database, 334
 bootsd command, 71
 Bourne shell, 49 to 50, 51, 227
 restricted, 45, 225
 box library routine, 269
 brk system call, 240
 BSD Source Compatibility Package, 39, 40
 bsearch library routine, 269
 byteorder library routine, 269
 bzero library routine, 269

C

C compilation system, directory for, 93, 342
 C compiler, 139 to 146, 170
 C program tools, 170 to ??
 C shell, 49 to 50, 51, 204
 C2conv command, 198
 C2unconv command, 198
 CACHEFS (cache file system), 78, 83
 cal command, 201
 calendar command, 201
 Calendar Manager, 4, 7
 calloc library routine, 269
 callrpc library routine, 269
 cancel command, 21, 118, 201, 216, 345
 capitalize command, 201
 captainfo command, 201
 cat command, 201
 catclose library routine, 269
 catgetmsg library routine, 175, 269
 catgets library routine, 175, 269
 catman command, 201
 catopen library routine, 269
 cb command, 170, 201
 cbc_crypt library routine, 270
 cbreak library routine, 270, 272
 cbrt library routine, 270
 cc command, 170, 201
 /ccs directory, *See* /usr/ccs directory
 cd command, 202
 cdc command, 168, 202
 CD-ROM devices
 improving performance, 78, 83
 install4x program and, 109 to 111
 managing, 22, 66 to 68
 naming conventions, 63
 SunOS release 5.4 requirement for, 25
 /cdrom directory, 66, 97
 CD-ROM file system (HSFS), 66, 82, 97
 ceil library routine, 270

cfgetispeed library routine, 270
 cfgetospeed library routine, 270
 cflow command, 170, 202
 cfree library routine, 270
 cfsetispeed library routine, 270
 cfsetospeed library routine, 270
 changing
 groups, 53
 NIS+ information, 20
 passwords, 44, 46
 shells, 49 to 50
 system run levels, 72 to 74
 users, 53
 character sets, 173
 chargefee command, 202
 chdir system call, 240
 check command, 118
 CHECK library routine, 265
 checkeq command, 202
 checking
 See also information reporting
 file systems, 84, 100 to 101
 port monitor status, 123
 printer status, 118, 345
 software package installation, 13, 167
 checknr command, 202
 chfn command, 202
 chgrp command, 202
 chkey command, 202
 chmod command, 201, 202
 chmod system call, 240
 chown command, 203
 chown system call, 240
 chroot library routine, 175, 203
 chroot system call, 240
 chrtbl command, 203
 chsh command, 203
 circle library routine, 270
 ckpacct command, 203
 /class directory, 342
 clear command, 203
 clear library routine, 270
 clear_colormap command, 203
 clear_function command, 203
 clearerr library routine, 270
 clearing inodes, 84
 clearok library routine, 270
 click command, 203
 clients
 See also Host Manager
 (Administration Tool); NIS
 (Network Information
 Services), clients
 diskless/dataless, 107 to ??
 clnt_broadcast library routine, 270
 clnt_call library routine, 270
 clnt_control library routine, 270
 clnt_create library routine, 270, 271
 clnt_destroy library routine, 270
 clnt_dg_create library routine, 271
 clnt_freeres library routine, 270
 clnt_geterr library routine, 270
 clnt_pcreateerror library
 routine, 270
 clnt_perrno library routine, 270
 clnt_perror library routine, 271
 clnt_raw_create library routine, 271
 clnt_spcreateerror library
 routine, 271
 clnt_sperrno library routine, 271
 clnt_sperror library routine, 271
 clnt_tli_create library routine, 271
 clnt_vc_create library routine, 271
 clntraw_create library routine, 271
 clnttcp_create library routine, 271
 clntudp_bufcreate library
 routine, 271
 clntudp_create library routine, 271
 clock command, 203
 clock library routine, 271
 close system call, 240
 closedir library routine, 271

closelog library routine, 271
 closepl library routine, 271
 clri command, 84, 203
 clrtobot library routine, 271
 clrtoeol library routine, 271
 clusters, 12, 25
 cmdtool command, 203
 cmp command, 203
 COFF, 169, 170
 col command, 203
 colcrt command, 203
 colldef command, 174, 203
 colltbl command, 174, 203
 coloredit command, 203
 colrm command, 203
 comb command, 168, 204
 comm command, 204
 command log file (UUCP), 129
 Command Tool (OpenWindows), 203
 command-line utilities, for software
 package administration, 12, 13
 commands
 See also specific commands
 changes in, 39
 See also Commands reference
 table
 device driver, 193
 file system, 84 to 86, 95 to 105
 local, 343
 messaging, 174
 NFS, 126
 NIS and, 19
 NIS+ and, 19
 printer, ?? to 120
 quick reference, 345 to 346
 Service Access Facility (SAF), 22, 122,
 123
 table of, 195 to 236
 Volume Management, 67
 Commands reference table, 195 to 236
 comment section, object files, 169
 community service messages, directory
 for, 341
 compatibility
 cross-functional hardware, 4, 5
 determining for applications, 165 to
 166
 SunOS release 4.x with release 5.4, 39
 to 41, 52, 135
 compatibility packages, 39 to 41, 135
 compilers, 139 to 146, 169
 compound library routine, 271
 compress command, 204
 concurrency, 8, 190
 config command, 71, 178, 180, 204
 Config file, 129
 configuration
 See also device configuration; system
 configuration
 auto, 8
 kernel, 16 to 17
 new features, 24 to ??
 configuration files, *See* /etc directory
 connect system call, 240
 consistency checking, *See* checking
 cont library routine, 272
 conventions, notation, xxv
 conversion tables, for iconv
 command, 343
 convert4x program, 108, 115
 converting
 archives to random libraies, 169
 files, 35 to 37
 copyright file, 166
 copysign library routine, 272
 copywin library routine, 272
 Core System Support, 12
 cos library routine, 272
 cosh library routine, 272
 cp command, 204
 cpio command, 104 to 105
 bar command vs., 200
 changes in, 104 to 105, 204
 described, 103, 104 to 105

- pax command vs., 221
 - paxcpio command vs., 221
 - support for, 101
- cpio system file, 334
- cpp command, 168, 204
- crash command, 204
- /crash directory, 341
- creat system call, 241
- creating
 - file systems, 84, 100
 - software packages, 167
- crmode library routine, 272
- cron command, 204, 214, 341
- /cron directory, 341
- /cron.d directory, 339
- crontab command, 204
- crontab system file, 334
- cross-functional compatibility, 4
- crtplot command, 204
- crypt command, 204
- _crypt library routine, 265
- crypt library routine, 265, 272
- cscope command, 170
- csh command, 49 to 50, 51, 204
- .cshrc file, 51
- csplit command, 204
- ctags command, 204
- ctermid library routine, 272
- ctime library routine, 273
- ctrace command, 170, 204
- cu command, 204
- curs_set library routine, 273
- cuserid library routine, 273
- customizing
 - kernel, 71
 - man command search path, 57 to 58
 - port monitors, 123
 - user environments, 51 to 52
- cut and paste, 6
- cut command, 205
- cv_broadcast library routine, 273

- cv_create library routine, 273
- cv_destroy library routine, 273
- cv_enumerate library routine, 273
- cv_notify library routine, 273
- cv_send library routine, 273
- cv_wait library routine, 274
- cv_waiters library routine, 274
- cxref command, 170, 205
- cylinder group maps, 14
- cylinder groups, 14

D

- daemons, quick reference, 347
- data access, common, 79
- Data Link Provider Interface (DLPI), 172
- data transfer
 - restoring, 32 to 37
 - saving, 25 to 31
- data types, expanded, 9
- databases
 - architecture-independent, 344
 - bootparams, 334
 - gettytab, 335
 - hosts, 335
 - inetd.conf, 335
 - locale, 335, 343
 - monetary, 174
 - netmasks, 335
 - networks, 335
 - NIS+, 341
 - phones, 336
 - printcap, 20, 37, 78, 117, 118, 119, 336, 347
 - printer subsystem, 343
 - protocols, 336
 - publickey, 336
 - rpc, 336
 - software package management utility
 - maintained, 94, 341
 - spell command, 342, 343, 344
 - termcap, 78, 337
 - terminfo, 78, 117, 119, 337, 344, 347
 - vgrindefs, 337

whatis, 58, 199
 windex, 58, 199
 yp, 337, 342

dataless clients, *See* diskless/dataless clients

date command, 174, 205

dbconfig command, 205

dbm_clearerr library routine, 274

dbm_close library routine, 274

dbm_delete library routine, 274

dbm_error library routine, 274

dbm_fetch library routine, 274, 279

dbm_firstkey library routine, 274, 280

dbm_nextkey library routine, 274, 298

dbm_open library routine, 274

dbm_store library routine, 274, 313

dbmclose library routine, 274

dbminit library routine, 274

dbx command, 154, 170, 205

dbxtool command, 154, 170, 205

dc command, 205

-dcheck command, 205

dd command, 101, 103, 205

DDI (Device Driver Interface), 9, 186, 187, 192

DDI/DKI (Device Driver Interface/Driver Kernel Interface), 9, 186, 187 to 188, 193

DDI/DKI compliant
 defined, 187

ddi_create_minor_node
 command, 181

DDI-compliant device drivers,
 defined, 187

Debugger (SPARCworks), 154

debuggers, 84, 154 to 156, 170
See also dbx command; dbxtool
 command; adb command;
 kadb command; truss
 command

decimal_to_double library
 routine, 274

decimal_to_extended library
 routine, 274

decimal_to_floating library
 routine, 274

decimal_to_single library
 routine, 274

def_prog_mode library routine, 274, 306

def_shell_mode library routine, 274
 /default directory, *See* /etc/default
 directory

defaults
 automounting, 18, 81
 directories, 79 to 81
 directory for, 45, 88, 340
 file systems, 79 to 81, 86
 root access, 45
 shell, 49 to 50
 swap device, 79, 83
 Volume Management file system, 79
 window system, 52

defaults_from_input command, 205

defaults_merge command, 205

defaults_to_indentpro
 command, 205

defaults_to_mailrc command, 205

defaultsed command, 205

del_curterm library routine, 274

delay_output library routine, 274

delch library routine, 275

delete library routine, 275

deleteln library routine, 275

deleting, *See* removing

delta command, 168, 206

delwin library routine, 275
 /demo directory, 342

deroff command, 206

des command, 206

des_crypt library routine, 275

des_setparity library routine, 275

DeskSet, 4, 6 to 7

desktop integration services, 6
 /dev directory

changes in, 14, 61, 72, 87, 339
 described, 80, 183, 339
 /`devices` directory and, 182
 reconfiguration boot and, 181
 /`dev/dsk` directory, 87, 183, 339
 /`dev/ksyms` file, 155
 /`dev/pts` directory, 87, 339
 /`dev/rdisk` directory, 87, 183, 339
 /`dev/rmt` directory, 87, 339
 /`dev/sad` directory, 87, 339
 /`dev/sd1g` command, 72
 /`dev/term` directory, 87, 339
`dev_info` nodes, 192
 Developer System Support, 12
 developers, 137 to 193
 Binary Compatibility Package
 and, 41, 165 to 166
 compilers, 139 to 146
 debuggers, 84, 154 to 156
 device configuration, 181 to 184
 device drivers, 185 to 193
 internationalization, 173 to 175
 linkers, 146 to 153
 networking features, 171 to 172
 Solaris features for, 8 to 9
 STREAMS, 191 to 193
 system configuration, 177 to 180
 tools and resources, 157 to ??
 transition information overview, 137
 device administration, 61 to 68
 adding devices, 65 to 66, 181, 193
 automatic loading, 9
 improving performance, 78, 83
 information reporting, 7, 63 to 65, 188
 to 189
 loading devices, 9, 17, 178, 193
 naming conventions, 14, 61 to 63, 181
 to 184
 removing devices, 180, 193
 unloading devices, 17, 178, 193
 Volume Management for, 22, 66 to 68,
 79
 device configuration, 37, 88, 181 to 184
 See also metadvice configuration
 information
 Device Driver Interface (DDI), 9, 187, 192
 Device Driver Interface/Driver Kernel
 Interface (DDI/DKI), 9, 186, 187
 to 188, 193
 device drivers, 185 to 193
 See also device administration; device
 configuration; device
 naming; specific devices
 changes in, 185
 commands, 193
 DDI-compliant, 187
 directories, 17, 94
 interface compatibility, 9
 interface types, 9, 186 to 188
 leaf drivers, 192, 193
 nexus drivers, 193
 porting considerations, 189 to 191
 STREAMS, 191 to 193
 device naming
 CD-ROMS, 63
 changes in, 14, 61 to 63
 developer's perspective, 181 to 184
 disks, 62, 63, 183
 instance names, 183 to 184
 logical names, 183
 physical names, 182
 tape drives, 32, 63
 device special file system (SPECFS), 82
 device tree, 181, 192 to 193
 /`devices` directory, 181
`devinfo` command, 63, 65, 188 to 189, 206
`devinfo` tree, 192
`devnm` command, 206
`df` command, 63, 64, 84, 98, 206
`dfmounts` command, 97, 346
 DFS (distributed file system)
 administration, 126
 /`dfs` directory, *See* /`etc/dfs` directory
`dfshares` command, 97, 346
`dgettext` library routine, 175
 /`dict` directory, 342
`diff` command, 206

diff3 command, 206

diffmk command, 206

dir system file, 334

dircmp command, 206

directories

- See also specific directories*
- changes in, 78, 87 to 89
- default, 79 to 81
- device driver, 17
- kernel module, 17, 178
- monitoring, 98
- names, 78
- path name generation for file systems, 84
- quick reference, 347
- remote, automounting, 17 to 18
- searching at link time, 149 to 150

dirname command, 206

dis command, 169, 207

disabling port monitor services, 123

discover4x program, 108

disk devices

- directories for, 87
- information reporting, 63 to 65, 84, 98
- names of disks attached to system, 26
- naming conventions, 62, 63, 183
- partition information, saving, 26 to 27

disk slices, 13 to 14

diskette devices, managing, 22, 66 to 68

diskette file system, automounting of, 66, 97

diskless/dataless clients, Solaris 2.4 server to support SunOS release 4.x, 107 to ??

diskusg command, 207

dispadmin command, 342

distributed file system (DFS)

- administration, 126

dkctl command, 207

DKI (Driver Kernel Interface), 9, 186

dkinfo command, 27, 63, 64, 207

dkio ioctls, 157, 158

dlclose library routine, 275

dlopen library routine, 275

DLPI (Data Link Provider Interface), 172

dlsym library routine, 275

dmesg command, 207

dn_comp library routine, 275

dn_expand library routine, 275

dname command, 207

DNS (domain name system)

- described, 131, 132
- NIS+ (Network Information Services Plus) vs., 132 to 133
- SunOS release 5.x and, 19, 31

document tools, using, 55 to 56

dodisk command, 207

dollar sign (library routine), xxv

domain name system. *See* DNS (domain name system)

domainname command, 207

dorfs command, 207

dos2unix command, 207

double_to_decimal library routine, 275

doupdate library routine, 275

drag and drop, 6

draino library routine, 275

drand48 library routine, 275, 287, 303, 309

Driver Kernel Interface (DKI), 9, 186

- See also* DDI/DKI (Device Driver Interface/Driver Kernel Interface)

drivers, *See* device drivers

- /drv directory, 17, 94
- drvconfig program, 181
- /disk directory, 87, 183, 339

du command, 63, 64, 98, 207

dumbplot command, 207

dump command

- See also* objdump command; ufsdump command
- changes to old, 101, 207

- link checks using, 151, 152
- new, 169
- new 218, xxi
- quick reference, 346
- dump system file, 334
- dumpfs command, 208
- dumpkeys command, 208
- dup system call, 241
- dup2 system call, 241
- dynamic kernel
 - See also* kernel
 - defined, 8, 16, 177 to 178
- dynamic linking, 9
- dysize library routine, 275

E

- e command, 208
- ecb_crypt library routine, 275
- echo command, 208
- echo library routine, 275
- echochar library routine, 275
- econvert library routine, 275
- ecvt library routine, 275
- ed command, 208
- edata library routine, 275
- edit command, 208
- edquota command, 208
- eeprom command, 208
- EFT (Extended Fundamental Types), 26
- egrep command, 208
- eject command, 208
- ELF (executable and linking format) files
 - compiler and, 139
 - kernel modules to run, 94
 - linker and, 146
- e-mail, *See* mail
- enabling port monitor services, 123
- encrypt library routine, 275
- end library routine, 275
- End User System Support, 12
- endac library routine, 275
- endexportent library routine, 276
- endfsent library routine, 276
- endgraent library routine, 276
- endgrent library routine, 276
- endhostent library routine, 276
- endmntent library routine, 276
- endnetent library routine, 276
- endnetgrent library routine, 276
- endprotoent library routine, 276
- endpwaent library routine, 276
- endpwent library routine, 276
- endrpcent library routine, 276
- endservent library routine, 276
- endttyent library routine, 276
- endusershell library routine, 276
- endwin library routine, 276
- enroll command, 208
- Entire Distribution, 12
- env command, 208, 221
- environ system file, 334
- environment initialization files
 - converting, 32
 - customizing user environments, 51 to 52
- environments. *See* Solaris operating environment; user environment administration
- eqn command, 208
- erand48 library routine, 277
- erase library routine, 277
- erasechar library routine, 277
- erf library routine, 277
- erfc library routine, 277
- errno library routine, 277
- errno values, 238
- error command, 168, 208
- error messages
 - file system commands, 85
 - Volume Management, 67
- /etc directory
 - See also* /usr/etc directory

changes in, 78, 86, 87 to 89, 339 to 340
 described, 80, 81, 87, 339
 /etc/.login file, 51
 /etc/acct directory, 339
 /etc/aliases file, 334, 347
 /etc/config command, 71, 178, 180, 204
 /etc/cron.d directory, 339
 /etc/default directory, 45, 88, 340
 /etc/default/fs file, 86
 /etc/default/login file, 45
 /etc/default/passwd file, 45
 /etc/default/su file, 45
 /etc/dfs directory, 340
 /etc/dfs/dfstab file, 36, 99
 /etc/dfs/fstype file, 86
 /etc/dfs/sharetab file, 266, 347
 /etc/exports file, 36, 99, 266, 334, 347
 /etc/fs directory, 86, 340
 /etc/fstab file
 described, 27, 335, 347
 /etc/vfstab vs., 33, 35, 36, 89, 347
 transferring information, 27 to 28
 /etc/group file, 36, 335
 /etc/inet directory, 88, 340
 /etc/init.d scripts, 223, 340
 /etc/inittab file, 71, 72
 /etc/lib directory, 340
 /etc/lp directory, 117, 21, 78, 88, 118, 340
 /etc/lp/printers directory, 119
 /etc/mail directory, 340
 /etc/mail/aliases file, 347
 /etc/mail/sendmail.cf file, 37
 /etc/mnttab file, 347
 /etc/mstab file, 335, 347
 /etc/net directory, 340
 /etc/netgroup file, 36, 335
 /etc/opt directory, 88, 94, 340
 /etc/passwd file, 36, 44, 335, 347
 /etc/printcap database, replacement
 of, 20, 37, 78, 117, 118, 119, 336,
 347
 /etc/profile file, 51
 /etc/rc scripts, 71, 73, 88, 180, 223, 340,
 347
 See also /sbin/rc scripts
 /etc/rc.boot script, 71, 73, 88, 180, 223
 /etc/rc.local script, 71, 73, 88, 180,
 223, 347
 /etc/rc.single script, 71, 88, 180
 /etc/rcn scripts, 71, 88, 180
 /etc/rcn.d scripts, 88, 180, 340, 347
 /etc/rcS script, 71, 88, 180
 /etc/rcS.d script, 88, 340, 347
 /etc/rmmount.conf file, 68
 /etc/saf directory, 88, 340
 /etc/sendmail.cf file, 37
 /etc/shadow file, 36, 44, 92, 347
 /etc/skel directory, 51, 340
 /etc/sm directory, 340
 /etc/sm.bak directory, 340
 /etc/system file
 described, 66, 71
 moddir variable, 17
 described, 179, 180
 moddir variable, 178
 /etc/tm directory, 340
 /etc/ttytab file, 37, 337
 /etc/uucp directory, 36, 127, 340
 /etc/uucp/Config file, 127, 129
 /etc/uucp/Grades file, 127
 /etc/uucp/Limits file, 127, 128
 /etc/vfstab file
 described, 89, 347
 example, 92
 fields, 89 to 92
 merging /etc/fstab file into, 27 to
 28, 33, 35, 36
 specifying file systems in, 97
 /etc/vold.conf file, 68
 /etc/xtab file, 337, 347
 etext library routine, 277
 ether library routine, 277
 ether_aton library routine, 277

ether_hostton library routine, 277
ether_line library routine, 277
ether_ntoa library routine, 277
ether_ntohost library routine, 277
etherd command, 208
etherfind command, 208
Ethernet drivers, 172
ethers system file, 334
EUC (extended UNIX code), 173
ex command, 208
ex editor, backup files directory, 341
exc_bound library routine, 277
exc_handle library routine, 277
exc_notify library routine, 277
exc_on_exit library routine, 277
exc_raise library routine, 277
exc_unhandle library routine, 277
exc_uniqpatt library routine, 277
/exec directory, 94
execl library routine, 277
execle library routine, 277
execlp library routine, 277
executable and linking format files, *See*
 ELF (executable and linking
 format) files
executables, building, 148 to 149
execv library routine, 278
execve system call, 241
execvp library routine, 278
exit library routine, 278
_exit system call, 239
exp library routine, 278
exp10 library routine, 278
exp2 library routine, 278
expand command, 208
expm1 library routine, 278
/export directory, 107, 340
/export/home directory, 200
exported file systems, *See* shared file
 systems

exportent library routine, 279
exportfs command, 97, 99, 208, 346
exports file, 36, 99, 266, 334, 347
expr command, 208
exstr command, 169
Extended Fundamental Types (EFT), 26
extended UNIX code (EUC), 173
extended_to_decimal library
 routine, 279
extract_files command, 208
extract_patch command, 208
extract_unbundled command, 208

F

fabs library routine, 279
FACE package, directories for, 343, 344
false command, 209
fastboot command, 70, 75, 209
fasthalt command, 72, 74, 75, 209
fctab system file, 334
fchdir system call, 241
fchmod system call, 242
fchown system call, 242
fchroot system call, 242
fclose library routine, 276, 279
fcntl system call, 242, 252
fcntl system file, 334
fconvert library routine, 279
fcvt library routine, 279
fdformat command, 209
FDFS (file descriptor file system), 79, 82
fdopen library routine, 279
feof library routine, 279
ferror library routine, 279
fetch library routine, 279
ff command, 84
fflush library routine, 279
ffs library routine, 279
fgetc library routine, 279
fgetgraent library routine, 279

fgetgrent library routine, 279
 fgetpwaent library routine, 279
 fgetpwent library routine, 279
 fgets library routine, 279
 fgrep command, 209
 FIFOFS (FIFO/pipe file system), 79, 83
 file command, 209
 file descriptor file system (FDFS), 79, 82
 File Manager, 4, 22, 66
 file system administration, *See* file systems
 file systems, 77 to 105
 See also files; *specific file systems*; *specific files*
 added, 79
 automounting, 17 to 18, 95 to 96, 97, 200
 backing up, 25 to 26, 30, 101 to 105
 CD-ROM devices with, 66
 changes in, 78 to 79
 format, 26
 layout, 339 to 344
 location, 78
 names, 35, 78
 checking, 84, 100 to 101
 commands
 changes in, 95
 described, 84 to 86
 locations, 86
 syntax, 85 to 86
 using, 95 to 105
 creating, 84, 100
 cylinder groups and, 14
 debugger, 84
 default, 79 to 81, 86
 disk slices and, 13 to 14
 diskette devices with, 66
 error message, 85
 exported, *See* shared file systems
 image copy of, 85
 kernel modules for implementing, 94
 labels, 84, 101, 214
 listing systems to save, 28
 monitoring, 98
 mounting, 17 to 18, 36, 84, 95 to 97, 218, 250, 346
 automounting, 17 to 18, 95 to 96, 97, 200
 mount table, 347
 path name list generation, 84
 pseudo, 78 to 79, 82 to 83
 remote
 automounting, 17 to 18
 default type, 86
 mounting, 84, 95
 unmounting, 84
 restoring, 30, 33, 103 to 105
 saving information, 27 to 28
 shared, *See* shared file systems
 specifying in /etc/vfstab file, 97
 supported types, 82 to 84
 System V, 9, 84
 temporary, *See* temporary file systems
 type determination, 84
 unmounting, 84, 96, 97
 unsupported SVR4, 84
 utility overview, 7
 virtual architecture (VFS), 82 to 86
 file_to_decimal library routine, 279
 fileno library routine, 279
 files
 See also file systems; *specific files*
 backing up, 25 to 26, 30, 101 to 105
 converting, 35 to 37
 environment initialization, 32, 51 to 52
 information reporting, 84
 listing files to save, 28
 merging, 34 to 35
 monitoring, 98
 names, 35, 78, 84
 opening using file descriptors, 79
 pipe, pseudo file system for, 79
 quick reference, 347
 restoring, 30, 33, 103 to 105
 searching for, 98
 system, 333 to 337
 filio ioctl's, 157, 159
 filter library routine, 279
 filters, image, unsupported, 21

find command, 98, 209
 finger command, 209
 fingerd command, 209
 finite library routine, 279
 firstkey library routine, 280
 fixterm library routine, 280
 flash library routine, 280
 floatingpoint library routine, 280
 flock system call, 242
 floor library routine, 280
 /floppy directory, 66, 97
 flushing library routine, 280
 flusok library routine, 280
 FMLI scripts, binaries directory for, 343
 fmod library routine, 280
 fmt command, 209
 fmt_mail command, 209
 fold command, 209
 /font directory, 343
 fontedit command, 209
 fonts
 description files directory for
 troff, 343
 device-independent, 55
 fopen library routine, 280, 308, 309
 foption command, 209
 foreign name service requests, entry
 points for, 343
 fork system call, 242
 format command, 26, 209
 fp_class library routine, 280
 fpa_download command, 209
 fparel command, 209
 fpathconf system call, 242
 fpaversion command, 209
 fprintf library routine, 280
 fpurel command, 209
 fputc library routine, 280
 fputs library routine, 280
 fpuversion command, 209
 fread library routine, 280
 free library routine, 270, 280
 freopen library routine, 280
 frexp library routine, 280
 from command, 209
 /fs directory, 86, 87, 94, 340, 343
 fs system file, 334
 fscanf library routine, 280
 fsck command, 84, 100 to 101, 205, 210
 fsck_cdrom command, 210
 fsdb command, 84, 212
 fseek library routine, 280
 fsflush command, 233
 fsirand command, 210
 fspec system file, 335
 /fstab system file, *See* /etc/fstab file
 fstat system call, 242
 fstatfs system call, 243
 fstatvfs system call, 243
 fstyp command, 84, 208
 fsync system call, 243
 ftell library routine, 280
 ftime library routine, 280
 ftok library routine, 281
 ftp command, 210
 ftpd command, 210
 ftpusers system file, 335
 ftruncate system call, 243
 ftw library routine, 281
 fumount command, 210
 func_to_decimal library routine, 281
 fusage command, 210
 fuser command, 210
 fwrite library routine, 281
 fwtmp command, 210

G

/games directory, 342
 games, binaries and data directory, 342
 gamma library routine, 281
 garbagedlines library routine, 281

gcd library routine, 281
gconvert library routine, 281
gcore command, 210
gcvt library routine, 281
generic file system commands, 84 to 86
generic_args command, 210
get command, 168, 210
get_alarm command, 210
get_myaddress library routine, 283
get_selection command, 210
getacdir library routine, 281
getacflg library routine, 281
getacinfo library routine, 281
getacmin library routine, 281
getauditflagsbin library routine, 281
getauditflagschar library routine, 281
getaud system call, 243
getbegyx library routine, 281
getc library routine, 281
getcap library routine, 281
getch library routine, 281
getchar library routine, 281
getcwd library routine, 175, 282, 285
getdate library routine, 315
getdents system call, 243
getdirent system call, 243
getdomainname system call, 243
getdtablesize system call, 243
getegid system call, 243
getenv library routine, 282
geteuid system call, 243
getexportent library routine, 282
getexportopt library routine, 282
getfauditflags library routine, 282
getfsent library routine, 282
getfsfile library routine, 282
getfsspec library routine, 282
getfstype library routine, 282
getgid system call, 244
getgraent library routine, 282
getgranam library routine, 282
getgrent library routine, 282
getgrgid library routine, 282
getgrnam library routine, 282
getgroups system call, 244
gethostbyaddr library routine, 283
gethostbyname library routine, 283
gethostent library routine, 283
gethostid system call, 244
gethostname system call, 244
getitimer system call, 244
getlogin library routine, 283
getmaxyx library routine, 283
getmntent library routine, 283
getmsg system call, 244
getnetbyaddr library routine, 283
getnetbyname library routine, 283
getnetent library routine, 283
getnetgrent library routine, 283
getnetname library routine, 283
getopt command, 210
getopt library routine, 283
getoptcv command, 210
getopts command, 210
getpagesize system call, 244
getpass library routine, 283
getpeername system call, 244
getpgid system call, 244
getpgrp system call, 245
getpid system call, 245
getppid system call, 245
getpriority system call, 245
getprotobyname library routine, 283
getprotobynumber library routine, 283
getprotoent library routine, 284
getpublickey library routine, 284
getpw library routine, 284
getpwaent library routine, 284
getpwanam library routine, 284

getpwent library routine, 284
getpwnam library routine, 284
getpwuid library routine, 284
getrlimit system call, 243, 246, 255, 323
getrpcbyname library routine, 284
getrpcbynumber library routine, 284
getrpcport library routine, 284
getrusage system call, 246, 324
gets library routine, 284
getsecretkey library routine, 284
getservbyname library routine, 284
getservbyport library routine, 284
getservent library routine, 284
getsockname system call, 246
getsockopt system call, 246
getstr library routine, 284
getsubopt library routine, 284
getsyx library routine, 284
gettable command, 210
gettext library routine, 175, 284
gettimeofday system call, 246
gettmode library routine, 284
getttyent library routine, 285
getttynam library routine, 285
getty command, 210
gettytab database, 335
getuid system call, 246
getusershell library routine, 285
getut library routine, 175
getvfsany library routine, 282
getvfsent library routine, 282
getvfsfile library routine, 282
getw library routine, 285
getwd library routine, 285
getyx library routine, 285
gfxtool command, 210
gigipolot command, 211
glob command, 211
glossary, 349 to 352
gmtime library routine, 285
goto command, 211
gpconfig command, 211
gprof command, 170, 211
graph command, 211
graphical user interfaces (GUIs)
 See also OpenWindows
 Administration Tool, 18 to ??
 for installation, 4, 8
 Software Manager, 12, 13
 XView Windows Toolkit, 167
graphics libraries, 6
grep command, 211
group system file, 36, 335
group.adjunct system file, 335
groups command, 211
groups, administering, 8, 53 to ??
groupware productivity tools, 4
grpauth library routine, 285
grpck command, 211
gsignal library routine, 285
gtty library routine, 285
GUIs, *See* graphical user interfaces (GUIs)
gxttest command, 211

H

halfdelay library routine, 285
halt command, 70, 74, 76, 211
halting, *See* shutting down
has_ic library routine, 285
has_il library routine, 285
hashcheck command, 211
hashmake command, 211
hashstat command, 211
hasmntopt library routine, 285
hcreate library routine, 285
hdestroy library routine, 285
head command, 211
header files, directory for, 191
help command, 168, 211

Help files, for mailx command, 344
 help_open command, 211
 holidays system file, 335
 /home directory, 340
 /home file system, 18, 80, 81
 \$HOME/.login file, 51
 \$HOME/.profile file, 51
 \$HOME/.cshrc file, 51
 \$HOME/ksh_env file, 51
 host2netname library routine, 285
 hostid command, 211
 hostname command, 211
 hosttrfs command, 211
 hosts database, 335
 hosts.equiv system file, 335
 hp7221plot command, 211
 hpplot command, 211
 hsearch library routine, 285
 HSFS (CD-ROM file system), 66, 82, 97
 htable command, 211
 HUGE library routine, 265
 HUGE_VAL library routine, 265
 hypot library routine, 285

I

i386 command, 212
 iAPX286 command, 212
 icode command, 212
 Icon Edit tool (OpenWindows), 212
 iconedit command, 212
 iconv command, conversion tables, 343
 /iconv directory, 343
 id command, 212, 234
 ID data types, expanded, 9
 idload command, 212
 idlok library routine, 286
 ieee_flags library routine, 286
 ieee_functions library routine, 286
 ieee_handler library routine, 286
 ieee_retrospective library

routine, 286
 ifconfig command, 212
 ilogb library routine, 286
 image copy, file systems, 85
 image filters, unsupported, 21
 Image Tool, 4, 7
 imemtest command, 212
 implot command, 212
 in.comsat command, 212
 in.fingerd command, 212
 in.ftpd command, 212
 in.named command, 212, 218
 in.rexcd command, 224
 in.rexd command, 224
 in.rexecd command, 212
 in.rlogind command, 212, 224
 in.routed command, 212, 225
 in.rshd command, 212
 in.rwhod command, 212
 in.talkd command, 212
 in.telnetd command, 212
 in.tftpd command, 212
 in.tnamed command, 212
 in.uucpd command, 212
 inch library routine, 286
 /include directory, 93, 342
 include header files, directory for, 342
 indent command, 170, 213
 indent.pro system file, 335
 indentpro_to_defaults
 command, 213
 index library routine, 286
 indxbib command, 213
 /inet directory, 88, 340
 inet_lnaof library routine, 286
 inet_makeaddr library routine, 286
 inet_netof library routine, 286
 inet_network library routine, 286
 inet_ntoa library routine, 286
 inetboot command, 71, 179, 180

`inetd` command, 213
`inetd.conf` database, 335
infinity library routine, 286
`infocmp` command, 213
information reporting
 See also checking
 device administration, 7, 63 to 65, 188 to 189
 disk devices, 63 to 65, 84, 98
 file systems, 84, 100 to 101
 kernel modules, 17, 178
 software packages, 13, 167
`init` command
 changes in, 72, 213
 commands replaced by, 70, 209
 described, 71
 using, 72 to 74
`init.d` scripts, 223, 340
`initgroups` library routine, 286
initialization files, *See* environment initialization files
initialization scripts, *See* scripts
initialization states, changing, 72 to 74
`initscr` library routine, 286
`initstate` library routine, 287
`inittab` file, 71, 72
`inline` command, 170, 213
`innetgr` library routine, 287
inodes, clearing, 84
`input_from_defaults` command, 213
`insch` library routine, 287
`insert_brackets` command, 213
`insertln` library routine, 287
`insque` library routine, 287
`install` command, 213
`install4x` program, 108, 109 to 114
installation, 23 to 37
 install phase, 23, 24, 31
 new features, 24 to ??
 overview, 8, 23 to 24
 post-installation phase, 23, 24, 32 to 37
 pre-installation phase, 25 to 31
 backing up file systems, 25 to 26, 30
 listing files and file systems to save, 28
 listing system components to save, 27
 network installation order, 29
 overview, 23, 24
 saving disk partition information, 26 to 27
 saving file system information, 27 to 28
 saving metadevice configuration information, 28 to 29
 saving SunOS release 4.x system data, 30 to 31
 software packages, 12 to 13, 343
 `installboot` command, 70, 179, 213
 `installtxt` command, 174, 213
 instance device names, 183 to 184
 integrity checking, *See* checking
 `internat` system file, 335
internationalization, 173 to 175, 343
Internet services
 configuration information directory, 88, 340
 network services directory, 343
 security, 44
interoperability, 5
interprocess communication utilities, 7
`intr` command, 213
`intrflush` library routine, 287
`ioctl` requests, 157 to 159, 246
 STREAMS, 191 to 192
`iostat` command, 213
`ipalloc` library routine, 287
`ipallocald` command, 213
`ipcrm` command, 213
`ipcs` command, 213
`irint` library routine, 287
`isalnum` library routine, 287
`isalpha` library routine, 287

isascii library routine, 287
 isatty library routine, 287
 iscntrl library routine, 287
 isdigit library routine, 287
 isendwin library routine, 287
 isgraph library routine, 287
 isinf library routine, 287
 islower library routine, 287
 isnan library routine, 287
 isnormal library routine, 287
 isprint library routine, 287
 ispunct library routine, 287
 issecure library routine, 287
 isspace library routine, 287
 issubnormal library routine, 287
 isupper library routine, 288
 isxdigit library routine, 288
 iszero library routine, 288
 itom library routine, 288

J

j0 library routine, 288
 j1 library routine, 288
 jn library routine, 288
 job grading, 126, 128
 join command, 214
 jrand48 library routine, 288

K

kadb command, 154 to 155, 214
 Kerberos security, 47
 kernel

- architecture-specific, 25
- booting and, 71
- configuration, 16 to 17
- crash dump directory, 341
- customizing, 71
- debugging a live, 155
- dynamic, 8, 16, 177 to 178
- layout, 17, 178

modules

- directory search path, 17, 178, 340, 342
- information reporting, 17, 178
- loading, 17, 178
- location, 15, 17, 25
- unloading, 17, 178
- MT (multithreaded), 8, 190
- name of, 15
- status information, 17

 /kernel directory, 17, 79, 94, 178, 340, 342
 /kernel/drv directory, 17, 94
 /kernel/exec directory, 94
 /kernel/fs directory, 94
 /kernel/misc directory, 94
 /kernel/sched directory, 94
 /kernel/strmod directory, 94
 /kernel/sys directory, 94
 /kernel/unix directory, 25, 71, 94, 178
 key_decryptsession library routine, 288
 key_encryptsession library routine, 288
 key_gendes library routine, 288
 key_setsecret library routine, 288
 keyboard layout description tables, 344
 keyenvoy command, 214
 keylogin command, 214
 keylogout command, 214
 keyname library routine, 288
 keypad library routine, 288
 keyserv command, 214
 /keytables directory, 344
 keytables system file, 335
 kgmon command, 214
 kill command, 214
 kill system call, 247
 killchar library routine, 288
 killpg system call, 247
 klm_prot library routine, 288
 Korn shell, 49 to 50, 51

restricted, 45
ksh command, 49 to 50, 51
ksyms file, 155
/kvm directory, 342
kvm_close library routine, 288
kvm_getcmd library routine, 288
kvm_getproc library routine, 288
kvm_getu library routine, 288
kvm_nextproc library routine, 288
kvm_nlist library routine, 289
kvm_open library routine, 289
kvm_read library routine, 289
kvm_setproc library routine, 289
kvm_write library routine, 289

L

l3tol library routine, 175, 289
l64a library routine, 289
label library routine, 289
labelit command, 84, 101, 214
langinfo library routine, 175
language products, mount/installation
point for, 340
large organizations, advantages of Solaris
for, 6
last command, 214
lastcomm command, 214
lastlogin command, 214
LC_MONETARY, 174
lcong48 library routine, 289
ld command, 168, 214
ldaclose library routine, 289
ldahread library routine, 289
ldaopen library routine, 289
ldclose library routine, 289
ldconfig command, 214
ldd command, 214
ldexp library routine, 289
ldfcn library routine, 289
ldfhread library routine, 289
ldgetname library routine, 289
ldlinit library routine, 289
ldlitem library routine, 289
ldlread library routine, 289
ldlseek library routine, 289
ldnlseek library routine, 290
ldnrseek library routine, 290
ldnshread library routine, 290
ldnsseek library routine, 290
ldohseek library routine, 290
ldopen library routine, 290
ldrseek library routine, 290
ldshread library routine, 290
ldsseek library routine, 290
ldtbindex library routine, 290
ldtbread library routine, 290
ldtbseek library routine, 290
leaf drivers, 192, 193
leaf nodes, 192
leave command, 214
leaveok library routine, 290
lex command, 168, 214
lfind library routine, 290
lgamma library routine, 290
/lib directory, *See* /etc/lib directory;
/usr/ccs/lib directory; /usr/lib
directory; /usr/share/lib
directory
libc directory, 174, 175
libintl directory, 174, 175
/libp directory, 163, 343
libraries
changes in, 161 to 164
converting archives to random, 169
dynamic linking of, 9
lint, 139
names and locations, 162 to 164
networking, shared objects and, 162
profiled, 343
resource limits, 161 to 163
search path rules, 150
search path specification, 149

shared
 building, 148, 152
 changes in, 161
 version numbering, 150 to 151, 161
 table of routines, 263 to 332
 Library Routine reference table, 263 to 332
 libsocket directory, 172
 libw directory, 173, 175
 /libxpg directory, 175
 /libxpg2.a library, 174
 limits
 See also defaults
 resource, 161 to 163
 line command, 215
 line library routine, 290
 linemod library routine, 290
 link command, 215
 link system call, 247
 link system file, 335
 linkers, 146 to 153
 building executables, 148 to 149
 building shared libraries, 148, 152
 dynamic linking, 9
 examples, 151 to 153
 library search path rules, 150
 library search path specification, 149
 link editor option changes, 146 to 148
 version numbering, 150 to 151
 lint command, 139, 170, 215
 lint libraries, 139
 listen port monitor, 21, 22, 123, 124, 215
 listen system call, 247
 listing
 file names and statistics, 84
 file systems to save, 28
 mounted resources, 347
 shareable resources, 347
 shared file systems, 347
 software packages installed, 13
 ln command, 215
 loading device drivers, 9, 17, 178, 193
 loadkeys command, 215
 local commands directory, 343
 /local directory, 343
 /local.cshrc file, 51
 /local.login file, 51
 /local.profile file, 51
 localdtconv library routine, 290
 locale database, 335, 343
 localeconv library routine, 290
 localization databases, 343
 localtime library routine, 290, 320
 lockd command, 215, 225
 lockf library routine, 290
 locks
 condition variables, 190
 master, 190
 multithreaded-style, 190
 mutexes, 155, 190
 network lock daemon, 347
 spooling lock files directory, 341
 /locks directory, 341
 lockscreen command, 215
 LOFS (loopback file system), 82
 log files
 cron, 341
 Service Access Facility (SAF), 341
 system, 341
 uucp, 129, 342
 log library routine, 290
 log10 library routine, 291
 loglp library routine, 291
 log2 library routine, 291
 logb library routine, 291
 logger command, 215
 logical device names, 183
 login command, 215
 .login file, 51
 login file, 45
 login shells
 default home directory startup files, 51
 features, 50
 initialization files, 51

restricted, 45
 selecting default, 49 to 50
 logins, administering, 22, 45
 logname command, 215
 logname library routine, 175
 _longjmp library routine, 265
 longjmp library routine, 291
 longname library routine, 291
 look command, 215
 lookbib command, 215
 loopback file system (LOFS), 82
 lorder command, 168, 215
 lp command, 20, 118, 120, 216, 345
 /lp directory, *See* /etc/lp directory;
 /usr/lib/lp directory; /var/lp
 directory; /var/spool/lp
 directory
 lpadmin command, 119, 216
 lpc command, 117, 119, 216
 lpd command, 117, 216, 347
 lpmove command, 119
 lpq command, 21, 117, 118, 216, 345
 lpr command, 20, 117, 118, 216, 345
 lprm command, 21, 117, 118, 216, 345
 lprof command, 230
 lpsched command, 347
 lpstat command, 21, 118, 216, 345
 lpsystem command, 119
 lptest command, 216
 lrand48 library routine, 291
 ls command, 98, 216
 lsearch library routine, 291
 lseek system call, 247
 lstat system call, 247
 lsw command, 216
 lto13 library routine, 291
 lwp_checkstkset library routine, 291
 lwp_create library routine, 291
 lwp_ctxinit library routine, 291
 lwp_ctxmemget library routine, 291
 lwp_ctxmemset library routine, 291

lwp_ctxremove library routine, 292
 lwp_ctxset library routine, 292
 lwp_datastk library routine, 292
 lwp_destroy library routine, 292
 lwp_enumerate library routine, 292
 lwp_errstr library routine, 292
 lwp_fpset library routine, 292
 lwp_geterr library routine, 292
 lwp_getregs library routine, 292
 lwp_getstate library routine, 292
 lwp_join library routine, 292
 lwp_libcset library routine, 292
 lwp_newstk library routine, 292
 lwp_perror library routine, 292
 lwp_ping library routine, 292
 lwp_resched library routine, 292
 lwp_resume library routine, 292
 lwp_self library routine, 292
 lwp_setpri library routine, 292
 lwp_setregs library routine, 292
 lwp_setstkcache library routine, 292
 lwp_sleep library routine, 292
 lwp_stkcswset library routine, 292
 lwp_suspend library routine, 292
 lwp_yield library routine, 292

M

m4 command, 168, 216
 m68k command, 216
 mach command, 216
 macros
 kadb, 154 to 155
 nroff, 344
 troff, 344
 madd library routine, 292
 madvise library routine, 293
 magic system file, 335
 magnetic tape devices, *See* tape devices
 mail
 auxiliary programs directory, 343

configuration information
 directory, 340
 directory for, 341, 347
 multimedia, 4
 queued directory, 341
 using, 54 to 55

Mail command, 198

mail command, 54 to 55, 201, 216
 See also **rmail** command

/mail directory, *See* /etc/mail directory;
 /usr/lib/mail directory;
 /var/mail directory;
 /var/spool/mail directory

Mail Tool (OpenWindows), 216

mailrc_to_defaults command, 216

mailstat command, 216

mailtool interface, 55, 216

mailx command, 55, 198, 216, 344
 /mailx directory, 344

make command, 164, 168, 217

makedbm command, 217

makedev command, 217

MAKEDEV environment variable, 72

Makefiles, 164

makekey command, 216, 217

malloc library routine, 175, 293

malloc_debug library routine, 293

malloc_verify library routine, 293

mallocmap library routine, 293

man command, 57 to 58, 59, 217
 /man directory, 56 to 58

man pages
 directory organization changes, 56 to 57
 notation conventions, xxv
 whatis database, 58, 199
 windex database, 58, 199

man.cf files, 57

man68010 command, 217

man68020 command, 217

man68881version command, 217

MANPATH environment variable, 57

MANSECTS environment variable, 58

maps (NIS), *See* NIS (Network Information Services)

maps (NIS+), *See* NIS+ (Network Information Services Plus)

master lock, 190

master servers, *See* NIS (Network Information Services), master servers

matherr library routine, 293

max_normal library routine, 293

max_subnormal library routine, 293

maximums, *See* limits

mblen library routine, 293

mbstowcs library routine, 293

mbtowc library routine, 293

mcmp library routine, 293

mconnect command, 217

mcs command, 169

mctl system call, 247

mdiv library routine, 293

memalign library routine, 293

memccpy library routine, 293

memchr library routine, 293

memcmp library routine, 268, 293

memcntl system call, 247

memcpy library routine, 268, 293

memset library routine, 269, 293

menus, **sysadm** menu utility, 9 to 10

merging files, 34 to 35

mesg command, 217

message catalogs, 174

messages, spoken, 4

meta library routine, 293

metadb command, 29

metadevice configuration information, saving, 28 to 29

metastat command, 29

mfree library routine, 293

min library routine, 293

min_normal library routine, 293

min_subnormal library routine, 293
 mincore system call, 247
 /misc directory, 94
 mkdir command, 217
 mkdir system call, 248
 mkfifo system call, 248
 mkfile command, 217
 mkfs command, 84, 100, 217
 mknod command, 217
 mknod system call, 249
 mkproto command, 217
 mkstemp library routine, 293
 mkstr command, 217
 mktemp library routine, 294
 mktime library routine, 320
 mlock library routine, 294
 mlockall library routine, 294
 mmap system call, 249
 /mnt directory, 340
 mnttab file, 347
 moddebug macro, 155
 moddir variable, 17, 178
 modems, managing, 21 to 22, 121 to 124
 modf library routine, 294
 modifying, *See* changing
 modinfo command, 17, 178, 218
 modload command, 17, 178, 180, 193, 217
 modstat command, 218
 modules, *See* kernel; STREAMS
 modunload command, 17, 178, 193, 218
 mon_break library routine, 294
 mon_cond_enter library routine, 294
 mon_create library routine, 294
 mon_destroy library routine, 294
 mon_enter library routine, 294
 mon_enumerate library routine, 294
 mon_exit library routine, 294
 mon_waiters library routine, 294
 monacct command, 218
 moncontrol library routine, 294
 monetary databases, 174
 MONITOR library routine, 265
 monitor library routine, 295
 monitoring file systems, 98
 monstartup library routine, 295
 montbl command, 174
 more command, 218
 mount command, 18, 84, 96 to 97, 218, 346
 mount system call, 250
 mount_tfs command, 218
 mountall command, 84, 346
 mouted command, 218, 225
 mounting
 file systems, 17 to 18, 36, 84, 95 to 97,
 218, 250, 346
 automounting, 17 to 18, 95 to 96,
 97, 200
 mount table, 347
 listing mounted resources, 347
 remote resources, 84, 95
 mout library routine, 295
 move library routine, 295
 mprotect system call, 250
 /mqueue directory, 341
 mrand48 library routine, 295
 msg_enumrecv library routine, 295
 msg_enumsend library routine, 295
 msg_recv library routine, 295
 MSG_RECVALL library routine, 265
 msg_reply library routine, 295
 msg_send library routine, 296
 msgctl system call, 250
 msgfmt command, 174, 213
 msgget system call, 250
 msgrcv system call, 251
 msgsnd system call, 251
 msub library routine, 296
 msync library routine, 296
 msync system call, 251
 MT (multithreaded) kernel, 8, 190
 mt command, 218

mtab system file, 335, 347
mtio ioctls, 157, 159
mtx library routine, 296
MT-style locks, 190
mult library routine, 296
multimedia mail, 4
multiple OS operation, 108
multithreaded (MT) kernel, 8, 190
multithreaded (MT), defined, 190
multiuser run levels, 73
multiuser systems, shutting down, 74 to 75
munlock library routine, 296
munlockall library routine, 296
munmap system call, 251
mutex macro, 155
mutexes, 155, 190
mv command, 218
mvaddch library routine, 296
mvaddstr library routine, 296
mvcur library routine, 296
mvdelch library routine, 296
mvgetch library routine, 296
mvgetstr library routine, 296
mvinch library routine, 297
mvinsch library routine, 297
mvprintw library routine, 297
mvscanw library routine, 297
mwaddch library routine, 297
mwaddstr library routine, 297
mwdelch library routine, 297
mwgetch library routine, 297
mwgetstr library routine, 297
mvwin library routine, 297
mvwinch library routine, 297
mvwinsch library routine, 297
mvwprintw library routine, 297
mvwscanw library routine, 297

N

name service request, foreign, entry points for, 343
Name Service Switch, 20, 132, 172
name services, *See* DNS (domain name system); NIS+ (Network Information Services Plus); NIS (Network Information Services)
named command, 218
NAMEFS (name file system), 79, 83
names
 device
 CD-ROMs, 63
 changes in, 14, 61 to 63
 developer's perspective, 181 to 184
 disks, 62, 63, 183
 instance, 183 to 184
 logical, 183
 physical, 182
 tape drives, 32, 63
 directories, 78
 disks attached to system, 26
 explicit, for opening files using file descriptors, 79
 file, 35, 78, 84
 kernel name, 15
 libraries, 162 to 164
napms library routine, 298
nawk command, 218
ncheck command, 84, 205, 218
ndbootd command, 218
neqn command, 218
/net directory, *See* /etc/net directory; /usr/net/servers directory
net_addr library routine, 298
netdir_getbyname library routine, 283
netgroup system file, 36, 335
netmasks database, 335
netname2host library routine, 298
netname2user library routine, 298
netrc system file, 335

netstat command, 218
/netsvc directory, 225, 343
network devices, administering, 21 to 22
Network Information Services Plus, *See*
NIS+ (Network Information
Services Plus)
Network Information Services, *See* NIS
(Network Information Services)
Network Interface Tap (NIT), 172
network lock daemon, 347
networking libraries, shared objects
and, 162
networks
backups across, 103
booting over, 71, 179, 180
configuration information directory
for transport-
independent, 340
order of installing Solaris 2.4 on, 29
service administration, 21 to 22, 125
to 129
Solaris 2.4 features, 171 to 172
networks database, 335
newaliases command, 218
newfs command, 100, 218
newgrp command, 218
newkey command, 218
newpad library routine, 298
/news directory, 341
newterm library routine, 298
newwin library routine, 298
nextafter library routine, 298
nextkey library routine, 298
nexus drivers, 193
nexus nodes, 193
NFS
automounting of file systems shared
through, 17 to 18
auxiliary programs and daemons
directory, 343
binder process, 347
command changes, 126
daemons, 343, 347
listing mounted, 346
listing shared (exported), 346
mount request server, 347
secure, 44
support for, 82
/nfs directory, 343, 347
nfsd command, 218, 347
nfsstat command, 218
nfssvc system call, 251
nfssys system call, 251
nice command, 219
nice library routine, 298
nint library routine, 298
NIS (Network Information Services)
automounting and, 18
clients, restoring system data for, 34
command and functions using, 19
master servers
restoring system data for, 34
saving system data for, 31
NIS+ migration, 134, 135
NIS+ support for, 19, 31, 135
NIS+ vs., 132, 134
SunOS release 5.x and, 19, 31
/nis directory, 341
nis prefix, 19
NIS+ (Network Information Services Plus)
advantages, 20
automounting and, 18
command and functions using, 19
database directory, 341
described, 4, 8, 19, 131, 132, 172
DNS vs., 132 to 133
installing on SunOS release 4.x
system, 135 to 136
NIS migration to, 134, 135
NIS support by, 19, 31, 135
NIS vs., 132, 134
planning an upgrade, 135
searching, 20
TCP/IP and, 125
updating, 20
NISBCP compatibility package, 135
nispaswd command, 46

NIT (Network Interface Tap), 172
nl command, 219
nl library routine, 298
nl_init library routine, 298
nl_langinfo library routine, 298
nlist library routine, 298
nlm_prot library routine, 298
nlsadmin command, 21, 219
nm command, 168, 219
nocbreak library routine, 298
nocrmode library routine, 298
nodelay library routine, 298
nodes, device tree, 192 to 193
noecho library routine, 298
nohup command, 219
nonl library routine, 298
nonstandard_arithmetic library routine, 299
noraw library routine, 299
notation conventions, xxv
notimeout library routine, 299
nrand84 library routine, 299
nroff command, 219, 343, 344
nslookup command, 219
nsquery command, 219
nsswitch.conf file, 172
/nterm directory, 344
ntohl library routine, 299
ntohs library routine, 299
nulladm command, 219

O

/oasys directory, 343
objdump command, 170
object code disassembler, for COFF, 169
od command, 219
/old directory, 93, 169, 343
old-analyze command, 219
old-cat command, 220
old-clocktool command, 220

old-compact command, 220
old-eyacc command, 220
old-filemerge command, 220
old-make command, 220
old-perfmon command, 220
old-prmail command, 220
old-pti command, 220
old-setkeys command, 220
old-sun3cvt command, 220
old-syslog command, 220
old-uncompact command, 220
old-vc command, 220
OLIT (OPEN LOOK Intrinsic Toolkit), 167
on command, 220
on_exit library routine, 299
OPEN LOOK Administration Tool, 4
OPEN LOOK Intrinsic Toolkit (OLIT), 167
open system call, 251
opendir library routine, 299
opening files, using file descriptors, 79
openlog library routine, 299
openpl library routine, 299
/openwin directory, 343
OpenWindows
Administration Tool, *See*
Administration Tool
clock command, 203
Command Tool, 203
Developer's Guide File Chooser vs.
XView File Chooser, 52
File Manager changes, 22, 66
Icon Edit tool, 212
indent command, 199
mail interface, 55
Mail Tool, 216
mount or installation point, 343
multiple displays, 198
Performance Meter tool, 221
PrintTool, 20
property sheets, 213

property window, 203, 205, 216, 227
 Shell Tool, 227
 SunView replaced by, 229
 Text Edit tool, 230
 version differences, 52
 xlock command, 215
 xset command, 226
 “Operation not applicable for FSType
 type” message, 85
 /opt directory, 88, 94, 340
 See also /var/opt directory
 /opt file system, 79, 80, 81
 /opt/sunwsprow directory, 340
 optarg library routine, 299
 optind library routine, 299
 optional packages, directories for, 340,
 341
 /options directory, 341
 order, of installing Solaris 2.4 on
 networks, 29
 overlay library routine, 300
 overview command, 220
 overwrite library routine, 300

P

pac command, 220
 pack command, 220, 233
 packages
 adding, 167
 administering, 343
 advantages, 12, 25
 compatibility packages, 39 to 41
 components, 166
 creating, 167
 defined, 12, 166
 developers and, 166 to 167
 information reporting, 13, 167
 installing, 12 to 13
 optional, 340, 341
 removing, 12 to 13, 167
 spooled, directory for, 341
 page command, 220
 pagesize command, 220
 passwd command, 46, 221
 passwd file, 36, 44, 45, 335, 347
 passwd.adjunct system file, 335
 passwd2des library routine, 300
 passwords
 aging, 44, 45
 changing, 44, 46
 location of, 36, 44, 335, 347
 paste command, 221
 path names
 See also search paths
 generating for file systems, 84
 pathconf system call, 251
 pause library routine, 300
 pax command, 221
 paxcpio command, 221
 pcat command, 221
 PCFS (PC file system), 82
 pclose library routine, 300
 pdpl1 command, 221
 pechochar library routine, 300
 percent sign (%), xxv
 perfmeter command, 221
 performance log file (uucp), 129
 Performance Meter tool
 (OpenWindows), 221
 perror library routine, 300
 personal productivity tools, 4, 6 to 7
 pg command, 221
 phones database, 336
 physical device names, 182
 PIC archives directory, 343
 /pics directory, 343
 ping command, 221
 pipe files, pseudo file system for, 79
 pipe system call, 251
 /pkg directory, 341
 pkgadd command, 12, 13, 167, 208
 pkgask command, 167
 pkgchk command, 13, 167
 pkginfo command, 13, 167

pkginfo file, 166
 pkgmk command, 167
 pkgparam command, 167
 pkgproto command, 167
 pkgrm command, 12, 13, 167
 pkgtrans command, 167
 plock library routine, 300
 plot command, 221
 plot library routine, 300
 plot system file, 336
 plottoa command, 221
 pmadm command, 21, 22, 122, 123, 124
 pmap_getmaps library routine, 300
 pmap_getport library routine, 284, 301
 pmap_rmtcall library routine, 301
 pmap_set library routine, 301
 pmap_unset library routine, 301
 pnoutrefresh library routine, 301
 pnp library routine, 301
 pod_exit library routine, 301
 pod_getexit library routine, 301
 pod_getmaxpri library routine, 301
 pod_getmaxsize library routine, 301
 pod_setexit library routine, 301
 pod_setmaxpri library routine, 301
 point library routine, 301
 point-to-point protocol (PPP), 126
 poll system call, 251
 popen library routine, 301
 port monitor services, administering, 21
 to 22, 122, 123
 port monitors
 administering, 21 to 22, 122, 123
 described, 122 to 123
 portability, 5
 porting applications, reasons for, 40
 porting device drivers, 189 to 191
 portmap command, 221
 PostScript filters, 55
 pound sign (#), xxv
 pow library routine, 302
 PPP (point-to-point protocol), 126
 pr command, 221
 praudit command, 221
 prctmp command, 221
 prdaily command, 221
 prefresh library routine, 302
 /preserve directory, 341
 print subsystem
 backend executables directory, 343
 configuration information
 directory, 340
 database directory, 343
 described, 20 to 21
 logging information directory, 341
 printcap database, *See* /etc/printcap
 database
 printenv command, 221
 printers, 117 to 120
 canceling print jobs, 21, 118, 201, 345
 changes in, 20, 117
 commands
 See also lp command
 changes in, 20
 using, 118 to 120
 configuration, 37, 88
 daemon file, 347
 network requests for services,
 administering and
 troubleshooting, 21
 setting up
 commands, 118 to 120, 345
 spool files directory, 341
 status checking, 118, 345
 printf library routine, 302
 PrintTool, 20
 printw library routine, 302
 priocntl command, 223, 342
 priocntl system call, 245, 255
 prioritizing, Solaris 2.4 installation for
 networks, 29
 /proc directory, 79
 /proc file system, 80, 81

process management
 pseudo file systems for, 79
 utilities, 7
 PROCFS (process access file system), 79, 82
 productivity tools, *See* DeskSet
 prof command, 168, 222
 prof library routine, 302
 profil library routine, 294, 295, 302
 profil system call, 251
 .profile file, 51
 profiled libraries, 343
 programming tools, *See* developers; tools (developer)
 PROM, booting from, 70 to 71
 prompts, xxv
 property sheets (OpenWindows), 213
 property window (OpenWindows), 203, 205, 216, 227
 proto system file, 336
 protocols database, 336
 prototype file, 166
 prs command, 169, 222
 prt command, 169, 222
 prtacct command, 222
 prtconf command, 65, 188, 189, 209
 prtvtoc command, 63, 64 to 65, 207
 ps command, 222, 345
 pscat (C/A/T) filters, 21, 55
 pseudo file systems, 78 to 79, 82 to 83
 pseudo terminal (pty) slave devices, directory for, 87, 339
 pseudo-device driver modules, directory for, 94
 psignal library routine, 303
 pstat command, 222, 345
 ptrace request values, 160
 ptrace system call, 252
 /pts directory, 87, 339
 ptx command, 222
 /pub director, 344
 publickey database, 336
 putchar library routine, 303
 putchar library routine, 303
 putenv library routine, 303
 putmntent library routine, 266
 putmsg system call, 252
 putp library routine, 303
 putpwent library routine, 303
 puts library routine, 303
 putw library routine, 303
 pwck command, 222
 pwd command, 222
 pwdauth library routine, 303
 pwdauthd command, 222

Q

qsort library routine, 303
 queuedefs system file, 336
 quick reference table, 345 to 347
 QuickCheck, 26
 quiet_nan library routine, 303
 quot command, 98, 222
 quota command, 222
 quotacheck command, 222
 quotactl system call, 252
 quotaoff command, 222
 quotaon command, 222

R

rand library routine, 303
 random library routine, 303
 ranlib command, 169, 223
 rarpd command, 223
 rasfilter8tol command, 223
 raster image filters, 21, 55
 rasterfile system file, 336
 rastrepl command, 223
 raw disk devices, directory for, 87, 183, 339
 raw library routine, 303

raw tape devices, directory for, 87, 339
 rc scripts, 71, 72 to 74, 78, 88, 92, 180, 223, 340, 347
 rc.boot script, 71, 73, 88, 180, 223
 rc.local script, 71, 73, 88, 180, 223, 347
 rc.single script, 71, 88, 180
 rcmd library routine, 303
 rcn scripts, 71, 73 to 74, 88, 180
 rcn.d scripts, 88, 340, 347
 rcp command, 223
 rcS script, 71, 74, 78, 88, 92, 180
 rcS.d script, 88, 340, 347
 rdate command, 223
 rdist command, 223
 /rdsk directory, 87, 183, 339
 rdump command, 223
 re_comp library routine, 305
 re_exec library routine, 305
 read system call, 252
 readdir library routine, 304
 readlink system call, 252
 readv system call, 253
 realloc library routine, 304
 realpath library routine, 304
 real-time priority scheduling, 4
 reboot command, 70, 74, 76, 223
 reboot system call, 253
 reconfiguration boot, 181
 recv system call, 253
 recvfrom system call, 253
 recvmsg system call, 253
 red command, 223
 refer command, 223
 /refer directory, 343
 refresh library routine, 305
 regcmp command, 170
 regexp library routine, 305
 registerrpc library routine, 305
 regular expression compiler, 170
 rehash command, 223
 reject command, 119
 rem_drv command, 180, 193
 remainder library routine, 305
 remexportent library routine, 305
 remote CD-ROM devices, install4x program and, 110 to 111
 remote file systems
 automounting, 17 to 18
 default type, 86
 mounting, 84, 95
 unmounting, 84
 remote procedure calls, administering, 22
 remote system file, 336
 remote systems
 Admintool, 18 to ??
 software package administration, 12 to 13
 remote tape drives, backing up to, 103
 remove_brackets command, 223
 removing
 devices, 180, 193
 groups, 53
 network devices, 21 to 22
 NIS+ information, 20
 port monitor services, 123
 port monitors, 123
 software packages, 12 to 13, 167
 users, 53
 remque library routine, 305
 rename system call, 253
 renice command, 223
 repquota command, 223
 res_init library routine, 305
 res_mkquery library routine, 305
 res_send library routine, 305
 reset command, 223
 reset_prog_mode library routine, 280, 305
 reset_shell_mode library routine, 305
 resetterm library routine, 305
 resetty library routine, 305
 resolv.conf system file, 336

resource limits, 161 to 163
 restartterm library routine, 306
 restore command, 32, 101, 223, 346
 restoring, 32 to 37
 See also cpio command; restore
 command; ufsrestore
 command
 file systems, 30, 33, 103 to 105
 SunOS release 4.x system data, 34 to
 37
 SunOS release 4.x user data, 32 to 33
 rev command, 223
 rewind library routine, 306
 rewinddir library routine, 306
 rex library routine, 306
 rexd command, 224
 rexec library routine, 306
 rexecd command, 224
 rfadmin command, 224
 rfmaster system file, 336
 rfpasswd command, 224
 /RFS file system, 78
 rfstart command, 224
 rfstop command, 224
 rfuaadmin command, 224
 rfudaemon command, 224
 rgb system file, 336
 .rhosts files, 44
 rhosts system file, 336
 .rhosts.equiv files, 44
 rindex library routine, 306
 ring_alarm command, 224
 rint library routine, 306
 ripoffline library routine, 306
 rksh command, 45
 rlogin command, 224
 rlogind command, 224
 rm command, 224
 rm_client command, 224
 rm_services command, 224
 rmail command, 224
 rmdel command, 169, 224
 rmdir command, 224
 rmdir system call, 253
 rmmount command, 67
 rmmount.conf file, 68
 rmnstat command, 224
 rmt command, 224
 /rmt directory, 87, 339
 rmtab system file, 336
 rnusers library routine, 306
 roffbib command, 225
 root access, defaults, 45
 root file system (/), 81, 339 to 342
 root file system (/), 28
 rootmenu system file, 336
 route command, 225
 routed command, 225
 rpc database, 336
 RPC, secure, 44, 47
 rpc.bootparamd command, 225
 rpc.etherd command, 225
 rpc.lockd command, 225, 347
 rpc.mountd command, 225, 347
 rpc.rexd command, 225
 rpc.rquotad command, 225
 rpc.rstatd command, 225
 rpc.rusersd command, 225
 rpc.rwalld command, 225
 rpc.showfhd command, 225
 rpc.sprayd command, 225
 rpc.statd command, 225
 rpc.user_agentd command, 225
 rpc.yppasswdd command, 225
 rpc.yppupdated command, 225
 rpc_broadcast library routine, 270
 rpc_call library routine, 269
 rpc_createerr library routine, 306
 rpcb_getaddr library routine, 301
 rpcb_getmaps library routine, 300
 rpcb_rmtcall library routine, 301

rpcb_set library routine, 301
 rpcb_unset library routine, 301
 rpcbind command, 221
 rpcgen command, 225
 rpcinfo command, 225
 rpow library routine, 306
 rquota library routine, 306
 rrestore command, 225
 rresvport library routine, 306
 rsh command, 45, 225
 rstat library routine, 306
 rtime library routine, 306
 run levels, *See* rc scripts
 runacct command, 225
 run-state transition operations, directory
 for, 88
 rup command, 225
 runtime command, 225
 rusage command, 226
 ruserok library routine, 306
 rusers command, 226
 rusers library routine, 306
 rwall command, 226
 rwall library routine, 306
 rwho command, 226

S

s.files, 165
 S5 (System V file system), 84
 sa command, 226
 /sa directory, 343
 SAC (Service Access Controller), 21 to 22
 sac command, 124
 sacadm command, 21, 22, 122, 123, 124
 sact command, 169, 226
 /sad directory, 87, 339
 /sadm directory, *See* /usr/sadm directory;
 /var/sadm directory
 /saf directory, *See* /etc/saf directory;
 /usr/lib/saf directory; /var/saf
 directory
 SAMECV library routine, 265
 SAMEMON library routine, 265
 SAMETHREAD library routine, 265
 sar command, 198, 222
 savecore command, 226
 saveterm library routine, 306
 savetty library routine, 306
 saving
 See also backing up
 disk partition information, 26 to 27
 file system information, 27 to 28
 metadevice configuration
 information, 28 to 29
 SunOS release 4.x system data, 30 to
 31
 /sbin directory
 See also /usr/sbin directory
 changes in, 78
 described, 81, 86, 92, 341
 /sbin/init command, 70, 71, 72 to 74
 /sbin/rc scripts, 72 to 74, 78, 88, 92
 See also /etc/rc scripts
 /sbin/rcn scripts, 73 to 74, 78, 88, 92
 /sbin/rcS scripts, 74, 88, 92
 /sbin/rcS scripts scripts, 78
 sbrk library routine, 175
 sbrk system call, 253
 scalability, 5, 9
 scalb library routine, 307
 scalbn library routine, 307
 scandir library routine, 307
 scanf library routine, 307
 scanw library routine, 307
 /sccs directory, 169
 SCCS (source code control system), 165
 sccs command, 169, 226
 /sccs directory, 165, 168
 sccs-admin command, 226
 sccs-cdc command, 226
 sccs-comb command, 226

sccs-delta command, 226
 sccsdiff command, 169, 226
 sccsfile system file, 336
 sccs-get command, 226
 sccs-help command, 226
 sccs-prs command, 226
 sccs-prt command, 226
 sccs-rmdel command, 226
 sccs-sact command, 226
 sccs-sccsdiff command, 226
 sccs-unget command, 226
 sccs-val command, 226
 /sched directory, 94
 scheduling, *See* Calendar Manager;
 process management
 scr_dump library routine, 307
 scr_init library routine, 307
 scr_restore library routine, 307
 screenblank command, 226
 screendump command, 226
 screenload command, 227
 script command, 227
 scripts
 accounting, 342
 FMLI, binaries directory for, 343
 package management, 343
 profile scripts for new user
 accounts, 340
 rc, 71, 72 to 74, 78, 88, 92, 180, 223,
 340, 347
 software package, 167
 spell command, 344
 system activity report package, 343
 scroll library routine, 308
 scrolldefaults command, 227
 scrollok library routine, 308
 SCSA (Sun common SCSI architecture),
 DDI only interfaces, 187
 SCSI disks
 multithreaded kernel and, 190
 naming conventions, 62
 Sun common SCSI architecture
 (SCSA), 187
 sdlg command, 72
 sdiff command, 227
 search paths
 kernel modules, 17, 178
 linker, 149 to 150
 searching
 for files, 98
 NIS+ tables, 20
 seconvert library routine, 308
 security, 43 to 47
 See also passwords
 ASET, 8, 46 to 47
 changes in, 43 to 47, 199
 features, 44 to ??
 Kerberos, 47
 level of, 8, 46 to 47
 overview, 43
 restricted shells, 45
 SunShield, 47
 unbundled, 47
 security log file (uucp), 129
 sed command, 227
 seed48 library routine, 308
 seekdir library routine, 308
 select system call, 253, 323
 selection_svc command, 227
 semctl system call, 253
 semget system call, 254
 semop system call, 254
 send system call, 254
 sendmail command, 224, 227
 sendmail.cf file, 37
 sendmsg system call, 254
 sendto system call, 254
 Serial Port Manager (Administration
 Tool), 121 to 122
 serial ports, controlling, 21 to 22, 37
 servers
 See also NIS (Network Information
 Services), master servers;
 Solaris 2.4 server

/servers directory, 343
Service Access Controller (SAC), 21 to 22
Service Access Facility (SAF), 21 to 22, 122 to 124
 auxiliary programs and daemons directory, 343
 commands, 22, 123
 described, 21 to 22, 122 to 124
 directory for, 88, 340
 logging and accounting file directory, 94, 341
 Service Access Controller (SAC) and, 21 to 22, 122
services system file, 336
set_alarm command, 227
set_curterm library routine, 308
set_term library routine, 309
set4 command, 227
setac library routine, 308
setaudit system call, 254
setaudit system call, 254
setbuf library routine, 308
setbuffer library routine, 308
setdomainname system call, 254
setegid library routine, 308
setegid system call, 255
seteuid library routine, 308
seteuid system call, 255
setexportent library routine, 308
setfsent library routine, 308
setgid library routine, 308, 309
setgid system call, 255
setgraent library routine, 308
setgrent library routine, 308
setgroups system call, 254
sethostent library routine, 309
sethostname system call, 254
setitimer system call, 254, 322, 323
_setjmp library routine, 265
setjmp library routine, 309
setkey library routine, 309
setkeys command, 227
setlinebuf library routine, 309
setlocale command, 175
setlocale library routine, 309
setlogmask library routine, 309
setmntent library routine, 309
setnetent library routine, 309
setnetgrent library routine, 309
setpgid system call, 254
setpgrp system call, 255
setpriority system call, 255
setprotoent library routine, 309
setpwaent library routine, 309
setpwent library routine, 309
setpwfile library routine, 309
setregid system call, 255
setreuid system call, 255
setrgid library routine, 309
setrlimit system call, 255
setrpcent library routine, 309
setruid library routine, 309
setscreg library routine, 309
setserverent library routine, 309
setsid command, 227
setsid system call, 255
setsockopt system call, 255
setstate library routine, 309
setsyx library routine, 309
setterm library routine, 310
settimeofday system call, 255
setttyent library routine, 310
setuid library routine, 310
setuid system call, 255, 309
setup_client command, 227
setup_exec command, 227
setupterm library routine, 310
setuseraudit system call, 255
setusershell library routine, 310
setvbuf library routine, 310
sfconvert library routine, 310

sgconvert library routine, 310
 sgetl system call, 256
 sh command, 49 to 50, 51, 227
 shadow file, 36, 44, 92, 347
 share command, 97, 99, 208, 346
 /share directory, *See* /usr/share directory
 /share file system, mounting, 96
 shareable resources, listing, 347
 shareall command, 99, 346
 shared file systems
 configuration information
 directory, 340
 default root of, 340
 described, 99
 listing, 347
 mounting, 96
 Solaris 2.4 server for SunOS release
 4.x diskless/dataless
 clients, 107 to ??
 transitioning, 28, 36
 shared libraries
 building, 148, 152
 changes in, 161
 version numbering, 150 to 151, 161
 shared objects, networking libraries
 and, 162
 sharetab file, 266, 347
 Shell Tool (OpenWindows), 227
 shells
 See also Bourne shell; C shell
 default home directory startup
 files, 51
 features, 50
 initialization files, 51
 restricted, 45
 selecting default, 49 to 50
 shelltool command, 227
 shift_lines command, 227
 shmctl system call, 256
 shmctl system call, 256
 shmdt system call, 256
 shmdt system call, 256
 shmget system call, 256
 showfh command, 227
 showfhd command, 225
 showmount command, 97, 227, 346
 shutacct command, 227
 shutdown command, 70, 74 to 75, 228
 shutdown system call, 256
 shutting down
 fastboot command, 70, 75, 209
 fasthalt command, 72, 74, 75, 209
 halt command, 70, 74, 76, 211
 reboot command, 70, 74, 76, 223
 shutdown command, 70, 74 to 75
 shutdown system call, 256
 sigaction library routine, 310
 sigaction system call, 256, 257
 sigaddset library routine, 310
 sigaltstack system call, 256
 sigblock system call, 256
 sigdelset library routine, 310
 sigemptyset library routine, 310
 sigfillset library routine, 310
 sigfpe library routine, 310
 siginterrupt library routine, 310
 sigismember library routine, 310
 siglongjmp library routine, 265, 310
 sigmask system call, 256
 signal library routine, 311
 signaling_nan library routine, 311
 signbit library routine, 311
 significand library routine, 311
 sigpause system call, 256
 sigpending system call, 256
 sigprocmask routine, 256
 sigprocmask system call, 256
 sigsetjmp library routine, 265, 311
 sigsetmask system call, 256
 sigsetops routines, 256
 sigstack system call, 256
 sigsuspend system call, 256
 sigvec system call, 257
 sin library routine, 311

single_precision library routine, 311
 single_to_decimal library routine, 311
 single-user run levels, 73, 74
 single-user systems, shutting down, 72
 sinh library routine, 311
 size command, 169, 228
 /skel directory, 51, 340
 skyversion command, 228
 sleep command, 190, 228
 sleep library routine, 311
 slices, *See* disk slices
 slk_clear library routine, 311
 slk_init library routine, 312
 slk_label library routine, 312
 slk_noutrefresh library routine, 312
 slk_refresh library routine, 312
 slk_restore library routine, 312
 slk_set library routine, 312
 slk_touch library routine, 312
 slow devices, improving performance, 78, 83
 /sm directory, 340
 sm system file, 336
 /sm.bak directory, 340
 sm_inter library routine, 312
 /snadm directory, 93, 344
 snoop command, 208
 socket system call, 257
 socketpair system call, 257
 sockets, 172, 257
 sockio ioctl's, 157, 159
 soelim command, 228
 software
 See also packages
 third-party and unbundled, 79, 81
 software clusters, *See* clusters
 software groups, 25
 Software Manager, *See* swmtool command
 software packages, *See* packages; software
 Solaris 2.4 DDI/DKI, 9, 186, 187 to 188, 193
 Solaris 2.4 DDI/DKI compliant, defined, 187
 Solaris 2.4 operating environment
 See also NIS+ (Network Information Services Plus); OpenWindows; Solaris operating environment
 additional documentation on, xxvii
 overview of major changes, 11 to 22
 Solaris 2.4 server, SunOS release 4.x
 diskless/dataless client support on, 107 to ??
 Solaris 2.4 SPARC DDI/DKI, 9, 186, 187 to 188, 193
 Solaris 2.x driver architecture, 192 to 193
 Solaris operating environment
 See also Solaris 2.4 operating environment
 advantages, 3 to 6
 compatibility, 4, 5
 developer features, 8 to 9
 features, 4, 6 to 9
 interoperability, 5
 large organizations and, 6
 portability, 5
 scalability, 5, 9
 SVR4 vs., 3 to 4, 6 to 10
 system administrator features, 7 to 8
 user features, 6 to 7
 Solstice DiskSuite, installation and, 28 to 29
 sort command, 228
 sortbib command, 228
 source code control system (SCCS), 165
 source code directory, 344
 space library routine, 312
 sparc command, 228
 SPARC DDI/DKI, 9, 186, 187 to 188, 193
 SPARC platforms, Solaris features and, 9
 SPARCServer Manager, installation and, 28 to 29
 SPARCworks Debugger, 154

SPECFs (device special file system), 82
 speed of devices, improving, 78, 83
 spell command, 228, 342, 343, 344
 /spell directory, 343, 344
 spellin command, 228
 spline command, 228
 split command, 228
 splN/splr pairs, 190
 spoken messages, 4
 spool directory, *See* /var/spool directory
 spray command, 228
 spray library routine, 312
 sprintf library routine, 312
 sputl system call, 257
 sqrt library routine, 312
 srand library routine, 312
 srand48 library routine, 312
 random library routine, 312
 /src directory, 344
 sscanf library routine, 312
 ssignal library routine, 311, 312
 standard processes, quick reference, 347
 standard_arithmetic library
 routine, 313
 standardization, 3, 4, 5
 standend library routine, 313
 stdout library routine, 313
 startup, 69 to 72
 startup command, 228
 stat system call, 257
 statfs system call, 257
 statically linked programs directory, 344
 statmon system file, 336
 status checking, *See* checking; information
 reporting
 status files, uucp, 342
 status monitor information directory, 340
 statvfs system call, 257
 STKTOP library routine, 265
 stopping system, *See* shutting down
 store library routine, 313
 strcasecmp library routine, 313
 strcat library routine, 313
 strchr library routine, 286, 313
 strcmp library routine, 313
 strcoll library routine, 313
 strcpy library routine, 313
 strcspn library routine, 313
 strdup library routine, 313
 streamio ioctls, 157, 159
 STREAMS, 191 to 193
 changes in, 191
 described, 8
 Ethernet drivers and, 172
 ioctl transparency, 191 to 192
 location of modules, 94
 pseudo file system for, 79
 pushing modules, 192
 Solaris 2.x driver architecture, 192 to
 193
 STREAMS Administrative Driver, entry
 point directory, 87, 339
 streamtab structure, 192
 strftime command, 174
 strftime library routine, 314
 string_to_decimal library
 routine, 314
 strings command, 229
 strings, extracting from source files, 169
 strioctl structure, 191
 strip command, 169, 229
 strlen library routine, 314
 /strmod directory, 94
 strncasecmp library routine, 314
 strncat library routine, 314
 strncmp library routine, 314
 strncpy library routine, 314
 strpbrk library routine, 315
 strptime library routine, 315
 strrchr library routine, 306, 315
 strspn library routine, 315

strtod library routine, 315
 strtok library routine, 315
 strtol library routine, 315
 strxfrm library routine, 315
 stty command, 223, 229
 stty library routine, 315
 stty_from_defaults command, 229
 su command, 229
 su file, 45
 subpad library routine, 315
 subwin library routine, 315
 sum command, 229
 Sun C compiler, 139 to 146
 sun command, 229
 Sun common SCSI architecture (SCSA),
 DDI only interfaces, 187
 Sun DDI (Sun Device Driver Interface), 9,
 187, 192
 sundiag command, 229
 suninstall command, 229
 SunOS release 4.x
 backing up file systems, 25 to 26, 30
 Backup Copilot, 26
 command changes, 39
 compatibility with SunOS release
 5.4, 39 to 41, 52
 device naming and, 14
 diskless/dataless clients, Solaris 2.4
 server support for, 107 to ??
 environment initialization file
 conversion, 32
 NIS+ installation on, 135 to 136
 overview of major changes in Solaris
 2.4, 11 to 22
 QuickCheck, 26
 restoring system data, 34 to 37
 restoring user data, 32 to 33
 saving disk partition information, 26
 to 27
 saving file system information, 27 to
 28
 saving metadvice configuration
 information, 28 to 29
 saving system data, 30 to 31
 tools, finding, 168 to ??
 work environment, using with Solaris
 2.4, 39 to 41, 52
 SunOS release 5.4
 Administration Tool, 8
 automounting, 17 to 18
 command changes, 39
 compatibility packages, 39 to 41, 52
 cross-functional compatibility, 4
 device naming, 14
 DNS and, 19, 31
 file systems, 15 to 16
 installation features, 24 to ??
 kernel configuration, 16 to 17
 kernel name in, 15
 NIS support by, 19, 31, 135
 print subsystem, 20 to 21, 117
 security, ?? to 47
 SunOS release 5.5
 Admintool, 18 to ??
 security, 43 to ??
 SunOS release 5.x
 DNS and, 31
 NIS and, 31
 SunOS/BSD Source Compatibility
 Package, 39, 40
 SunShield, 47
 SunView, 52, 342
 sunview command, 229
 sunview system file, 336
 /sunview1 directory, 342
 SUNWhinst, 108
 sv_acquire command, 229
 sv_release command, 229
 sv_xv_sel_svc command, 227
 svc_create library routine, 317
 svc_destroy library routine, 315
 svc_dg_create library routine, 317
 svc_fd_create library routine, 317
 svc_fds library routine, 315
 svc_fdset library routine, 315, 316
 svc_freeargs library routine, 316

svc_getargs library routine, 316
 svc_getcaller library routine, 316
 svc_getreq library routine, 316
 svc_getreqset library routine, 316
 svc_raw_create library routine, 317
 svc_reg library routine, 316
 svc_register library routine, 316
 svc_run library routine, 316
 svc_sendreply library routine, 316
 svc_tli_create library routine, 317
 svc_unreg library routine, 316
 svc_unregister library routine, 316
 svc_vc_create library routine, 317
 svcerr_auth library routine, 316
 svcerr_decode library routine, 316
 svcerr_noproc library routine, 316
 svcerr_noprogram library routine, 316
 svcerr_progvers library routine, 316
 svcerr_systemerr library routine, 316
 svcerr_weakauth library routine, 316
 svctcp_create library routine, 317
 svccraw_create library routine, 317
 svctcp_create library routine, 317
 svcudp_bufcreate library routine, 317
 svcudp_create library routine, 317
 svdtab system file, 336
SVR4
 interchanging data, 103 to 105
 Solaris vs., 3 to 4, 6 to 10
 unsupported file system types, 84
 svr4.make command, 164, 217
 swab library routine, 175, 317
 swap command, 222, 229, 345
 swap device, default, 79, 83
 SWAPFS (swap file system), 79, 83
 swapon command, 229
 swapon system call, 257
 swin command, 229
 switcher command, 229
 swmtool command
 add_services command vs., 198
 described, 12, 13
 extract_unbundled command
 vs., 208
 rm_services command vs., 224
 symlink system call, 257
 symorder command, 169, 229
 sync command, 229
 sync system call, 257
 syntax, file system commands, 85 to 86
 /sys directory, 78, 94, 95
 sys_siglist library routine, 317
 sys_unconfig command, 230
 sysadm menu utility, 9 to 10
 syscall system call, 257
 sysconf system call, 244, 246, 257
 sysdef command, 63, 206, 211
 sysinfo system call, 243, 244, 254
 syslog library routine, 317, 324
 syslog.conf system file, 336
 syslogd command, 230
 system accounting
 directories for, 339, 341
 utilities overview, 7
 system activity report package, 343
 system administration directories, 343
 system administrators
 See also Administration Tool; specific types of administration
 security procedures, 43 to 47
 Solaris features for, 7 to 8
 System Calls reference table, 237 to 261
 system command, 71
 system commands, directory for
 standard, 342
 system configuration
 changes in, 177 to 180
 default, 88
 restoring data, 34 to 37
 saving configuration files, 30 to 31
 system file, *See /etc/system file*
 System Files reference table, 333 to 337
 system header files, directory for, 191

system information utilities, 7
system initialization scripts, *See* rc scripts
system library routine, 317
system run levels, *See* rc scripts
system scheduling, *See* process management
System V file system (S5), 9, 84
System V Interface Definition, 161
systems system file, 336

T

t_accept library routine, 317
t_alloc library routine, 317
t_bind library routine, 317
t_close library routine, 317
t_connect library routine, 317
t_error library routine, 317
t_free library routine, 318
t_getinfo library routine, 318
t_getstate library routine, 318
t_listen library routine, 318
t_look library routine, 318
t_open library routine, 318
t_optmgmt library routine, 318
t_rcv library routine, 318
t_rcvconnect library routine, 318
t_rcvdis library routine, 318
t_rcvrel library routine, 318
t_rcvudata library routine, 318
t_rcvuderr library routine, 318
t_snd library routine, 318
t_snddis library routine, 318
t_sndrel library routine, 318
t_sndudata library routine, 318
t_sync library routine, 318
t_unbind library routine, 318
t300 command, 230
t300s command, 230
t4013 command, 230
t450 command, 230

tab setting escape sequences, directory for, 344
tabs command, 230
/tabset directory, 344
tail command, 230
talk command, 230
tan library routine, 318
tanh library routine, 318
tape devices
 backing up to remote, 103
 directory for raw, 87
 naming conventions, 32, 63
tar command
 bar command vs., 200
 changes affecting, 103 to 104, 230
 described, 103 to 104
 saving SunOS release 4.x system data using, 30 to 31
 support for, 101
 ustar command vs., 233
tar system file, 337
tbl command, 230
tcdrain library routine, 318
tcflow library routine, 318
tcflush library routine, 318
tcgetattr library routine, 318
tcgetpgrp library routine, 318
tcopy command, 230
tcov command, 170, 230
TCP/IP, 125
tcsendbreak library routine, 319
tcsetattr library routine, 319
tcsetpgrp library routine, 319
tdelete library routine, 319
tee command, 230
tek command, 230
tektool command, 230
tell system call, 257
telldir library routine, 319
telnet command, 230
template files, shell initialization, 51

tempnam library routine, 319
temporary file systems
 See also /tmp file system (TMPFS)
 directory for files not cleared during boot sequence, 342
 directory for spooled temporary files, 341
 transitioning, 28
 /term directory, 87, 339
term system file, 337
termcap database, 78, 337
terminal description files, directory for, 344
terminal devices, directory for, 87, 339
terminal tables, for nroff command, 344
terminals, managing, 21 to 22, 121 to 124
terminfo database, 78, 117, 119, 337, 344, 347
termio interface, 285, 315
termio ioctl's, 157, 159
termios ioctl's, 157, 159
test command, 230
TEX filters, 21, 55
Text Edit tool (OpenWindows), 230
textdomain library routine, 175, 319
textedit command, 230
textedit_filters command, 230
tfind library routine, 319
TFS (translucent file system), 78, 82
tfsd command, 230
tftp command, 230
 /tftpboot directory, 28
tgetent library routine, 319
tgetflag library routine, 319
tgetnum library routine, 319
tgetstr library routine, 320
tgoto library routine, 320
third-party software, file system for, 79, 81
thread macro, 154
threadlist macro, 155
tic command, 231
tigetflag library routine, 320
tigetnum library routine, 320
tigetstr library routine, 320
time command, 231
time library routine, 280, 320
time management, *See* Calendar Manager; process management
time zone information, 344
timegm library routine, 320
timelocal library routine, 320
times library routine, 320
timezone library routine, 320
tip command, 231
 /tm directory, 340
 /tmac directory, 344
 /tmp directory, 341
 /tmp file system (TMPFS)
 described, 80, 81
 transitioning, 28
 support for, 82
tmpfile library routine, 320
TMPFS, *See* /tmp file system (TMPFS)
tmpnam library routine, 321
toascii library routine, 321
toc system file, 337
_tolower library routine, 265
tolower library routine, 321
toolkits, 167
toolplaces command, 231
tools (developer), 157 to ??
 Binary Compatibility Package, 39, 40
 to 41, 165 to 166
 finding, 168 to ??
 ioctl requests, 157 to 159
 libraries, 161 to 164
 make command, 164
 OLIT, 167
 ptrace request values, 160
 SCCS, 165
 software packages, 166 to 167
 XView, 167

ToolTalk, 6
 touch command, 231
 touchline library routine, 321
 touchoverlap library routine, 321
 touchwin library routine, 321
 _toupper library routine, 265
 toupper library routine, 321
 tparm library routine, 321
 tput command, 231
 tputs library routine, 321
 tr command, 231
 trace command, 155, 231
 See also truss command
 traceoff library routine, 322
 traceon library routine, 322
 trademark files, directory for, 340
 traffic command, 231
 translate system file, 337
 translucent file system (TFS), 78, 82
 troff command, 231
 changes in, 21, 56, 119 to 120
 macro packages for, 344
 preprocessor for, 343
 troubleshooting
 printing service requests from
 network, 21
 Service Access Controller (SAC), 21
 TTY devices, 21
 trpt command, 231
 true command, 231
 truncate system call, 257
 truss command, 155 to 156, 170, 231
 tsearch library routine, 322
 tset command, 231
 tsort command, 169, 231
 tty command, 231
 TTY devices, administering and
 troubleshooting, 21
 ttyadm command, 21
 ttycom ioctl requests,
 unsupported, 158
 ttymon port monitor, 21, 22, 123, 124
 ttyname library routine, 322
 ttyslot library routine, 322
 ttysoftcar command, 231
 ttytab system file, 37, 337
 tunefs command, 231
 turnacct command, 231
 tvconfig command, 231
 twalk library routine, 322
 typeahead library routine, 322
 types system file, 337
 typographic conventions, in manual, xxv
 tzfile system file, 337
 tzset library routine, 322
 tzsetup command, 232
 tzsetwall library routine, 322

U

u370 command, 232
 u3b command, 232
 u3b15 command, 232
 u3b2 command, 232
 u3b5 command, 232
 uadmin system call, 253, 257
 ualarm library routine, 322
 /ucb directory, *See* /usr/ucb directory
 /ucbinclude directory, 344
 /ucplib directory, 344, 162, 163
 UFS, *See* Unix file systems (UFS)
 ufsboot command, 70 to 71, 179, 180
 ufsbootblk command, 70, 71
 ufsdump command
 commands replaced by, 101, 207, 223,
 346
 described, 101, 102 to 103
 ufsrestore command
 commands replaced by, 32, 101, 223,
 225
 described, 103
 quick reference, 346
 ul command, 232

- ulimit library routine, 322
- umask command, 232
- umask system call, 257
- umount command, 18, 84, 96, 232, 346
- umount system call, 257
- umount_tfs command, 232
- umountall command, 84, 97, 346
- unadv command, 232
- uname command, 199, 211, 216, 232
- uname system call, 258
- unbundled software, file system for, 79, 81
- uncompress command, 232
- unconfigure command, 232
- unctrl library routine, 322
- unexpand command, 232
- unget command, 169, 232
- ungetc library routine, 323
- ungetch library routine, 323
- unifdef command, 169, 232
- uniq command, 232
- units command, 232
- /unix directory, 25, 71, 94, 178
- Unix file systems (UFS)
 - See also* file systems
 - creating, 100
 - cylinder groups and, 14
 - disk slices and, 13 to 14
 - support for, 82
- unix kernel name, 15, 79, 180
 - See also* kernel
- Unix System V Release 4, *See* SVR4
- unix2dos command, 232
- Unix-to-Unix Copy (UUCP), 36, 126 to 129
 - See also* uucp command
- unlink command, 232
- unlink system call, 258
- unloading, device drivers, 17, 178, 193
- unmount system call, 258
- unmounting
 - file systems, 84, 96, 97
 - remote resources, 84
- unpack command, 233
- unshare command, 97, 346
- unwhiteout command, 233
- update command, 233
- updaters system file, 337
- uptime command, 233
- user access, controlling, 54
- User Account Manager (Administration Tool)
 - password changes and, 46
- user directories, default root of, 340
- user environment administration, 49 to 59
 - default shell selection, 49 to 50
 - document tool usage, 55 to 56
 - mail usage, 54 to 55
 - man command search path
 - customization, 57 to 58
 - man page directory organization changes, 56 to 57
 - SunOS 4.x work environment used with Solaris 2.4, 39 to 41, 52
 - user and group administration, 53 to ??
 - window system, 52
- user management utilities, 8
- user2netname library routine, 323
- useradd command, 53
- userdel command, 53
- usermod command, 53
- users command, 233
- users, Solaris features for, 6 to 7
- usleep library routine, 323
- /usr directory
 - described, 78
 - changes in, 93
 - described, 93, 341
- /usr file system
 - described, 80, 81
 - layout changes, 342 to 344
 - transitioning, 28
 - mount point for, 341
- /usr/4lib directory, 342

/usr/5bin directory, 93
 /usr/5include directory, 93
 /usr/5lib directory, 93, 162, 163
 /usr/bin directory
 described, 84
 SunOS release 4.x files moved to, 93
 described, 342
 generic file system administrative
 commands in, 84 to 85
 programming tools in, 168, 169
 quick reference, 347
 /usr/bin/csh command, 49 to 50, 51,
 204
 /usr/bin/ksh command, 49 to 50, 51
 /usr/bin/mail command, 54 to 55, 201,
 216
 /usr/bin/mailx command, 55, 198,
 216, 344
 /usr/bin/rksh command, 45
 /usr/bin/rsh command, 45, 225
 /usr/bin/sh command, 49 to 50, 51, 227
 /usr/bin/sunview1 directory, 342
 /usr/ccs directory, 93, 342
 /usr/ccs/bin directory, 165, 168 to 169,
 342
 /usr/ccs/bin/make command, 164,
 168, 217
 /usr/ccs/lib directory, 162, 163, 168,
 342
 /usr/ccs/lib/svr4.make
 command, 164, 217
 /usr/ccs/libp directory, 163
 /usr/demo directory, 342
 /usr/etc directory, 93
 See also /etc directory
 /usr/etc/biod command, 201, 347
 /usr/etc/nfsd command, 218, 347
 /usr/etc/rpc.lockd command, 225,
 347
 /usr/etc/rpc.mountd command, 225,
 347
 /usr/etc/ypbind command, 235, 347
 /usr/games directory, 342
 /usr/include directory, 93, 342
 /usr/kernel directory, *See* /kernel
 directory
 /usr/kvm directory, 342
 /usr/lib directory, 93, 162, 168, 169, 342
 /usr/lib/acct directory, 342
 /usr/lib/class directory, 342
 /usr/lib/dict directory, 342
 /usr/lib/font directory, 343
 /usr/lib/fs directory, 86, 87, 343
 /usr/lib/iconv directory, 343
 /usr/lib/libc directory, 174, 175
 /usr/lib/libintl directory, 174, 175
 /usr/lib/libp directory, 343
 /usr/lib/libw directory, 173, 175
 /usr/lib/locale directory, 335, 343
 /usr/lib/lp directory, 343
 /usr/lib/lp/lpsched command, 347
 /usr/lib/lpd command, 117, 216, 347
 /usr/lib/mail directory, 343
 /usr/lib/netsvc directory, 225, 343
 /usr/lib/netsvc/yp/ypbind
 process, 347, 235
 /usr/lib/nfs directory, 343
 /usr/lib/nfs/lockd daemon, 347
 /usr/lib/nfs/mountd server, 347
 /usr/lib/nfs/nfsd daemon, 347
 /usr/lib/pics directory, 343
 /usr/lib/refer directory, 343
 /usr/lib/rsh command, 45
 /usr/lib/sa directory, 343
 /usr/lib/saf directory, 343
 /usr/lib/spell directory, 343
 /usr/lib/uucp directory, 343
 /usr/local directory, 343
 /usr/net/servers directory, 343
 /usr/oasys directory, 343
 /usr/old directory, 93, 169, 343
 /usr/openwin directory, 343

/usr/opt file, 79
 /usr/sadm directory, 343
 /usr/sadm/bin directory, 343
 /usr/sadm/install directory, 343
 /usr/sbin directory, 86, 87, 93, 343, 347
 /usr/sbin/static directory, 344
 /usr/sccs directory, 165, 168, 169
 /usr/share directory, 344
 /usr/share file system
 mounting, 96
 /usr/share/lib directory, 344
 /usr/share/lib/keytables
 directory, 344
 /usr/share/lib/mailx directory, 344
 /usr/share/lib/nterm directory, 344
 /usr/share/lib/pub directory, 344
 /usr/share/lib/spell directory, 344
 /usr/share/lib/tabset
 directory, 344
 /usr/share/lib/terminfo
 database, 78, 117, 119, 337, 344,
 347
 /usr/share/lib/tmac directory, 344
 /usr/share/lib/zoneinfo
 directory, 344
 /usr/share/man directory, 56 to 58
 /usr/share/src directory, 344
 /usr/snadm directory, 93, 344
 /usr/ucb directory, 40, 168, 344
 /usr/ucb/df command, 63, 64, 84, 98,
 206
 /usr/ucb/du command, 63, 64, 98, 207
 /usr/ucb/mail command, 54 to 55, 201,
 216
 /usr/ucbinclude directory, 344
 /usr/ucblib directory, 162, 163, 344
 /usr/vmsys directory, 344
 /usr/xpg2bin directory, 93
 /usr/xpg2include directory, 93
 /usr/xpg2lib directory, 93, 164
 /usr/xpg2lib/libxpg directory, 175
 /usr/xpg2lib/libxpg2.a library, 174
 ustar command, 233
 ustat system call, 258
 utime library routine, 323
 utimes system call, 258
 utmp system file, 337
 uucheck command, 233
 uucico command, 233
 uucleanup command, 233
 UUCP (Unix-to-Unix Copy), 36, 126 to 129
 uucp command, 224, 233
 auxiliary programs and daemons
 directory, 343
 configuration information
 directory, 340
 file deposit directory, 341
 log and status file directory, 129, 342
 queued jobs directory, 341
 /uucp directory, *See* /etc/uucp directory;
 /usr/lib/uucp directory;
 /var/spool/uucp directory;
 /var/uucp directory
 uuencode command, 233
 uuencode system file, 337
 uuglist command, 127
 uulog command, 233
 uuname command, 233
 uupick command, 233
 uusched command, 233
 uusend command, 233
 uustat command, 233
 uuto command, 233
 uux command, 233
 uuxqt command, 234

V

vacation command, 234
 vadvise system call, 258
 val command, 169, 234
 valloc library routine, 323

/var directory, 119, 341
 changes in, 78
 described, 78, 80, 81
 changes in, 93 to 94, 341 to 342
 described, 93, 341
 /var/adm directory, 341
 /var/crash directory, 341
 /var/cron directory, 341
 /var/lp directory, 341
 /var/mail directory, 78, 94, 341, 347
 /var/news directory, 341
 /var/nis directory, 341
 /var/opt directory, 341
 /var/opt/*packagename* directory, 93
 /var/options directory, 341
 /var/preserve directory, 341
 /var/sadm directory, 94, 341
 /var/saf directory, 94, 341
 /var/spool directory, 119, 341
 /var/spool/cron directory, 341
 /var/spool/locks directory, 341
 /var/spool/lp directory, 119, 341
 /var/spool/mail directory, 78, 94, 347
 /var/spool/mqueue directory, 341
 /var/spool/pkg directory, 341
 /var/spool/uucp directory, 341
 /var/spool/uucppublic
 directory, 341
 /var/tmp directory, 342
 /var/uucp directory, 342
 /var/uucp/.Admin/account file, 129
 /var/uucp/.Admin/command file, 129
 /var/uucp/.Admin/perflog file, 129
 /var/uucp/.Admin/security file, 129
 /var/yp directory, 342
 varargs library routine, 323
 vax command, 234
 vc command, 169
 vedit command, 234
 version numbering, shared libraries, 150
 to 151, 161
 vfont system file, 337
 vfontinfo command, 234
 vfork system call, 258
 vfprintf library routine, 323
 VFS (virtual file-system architecture), 82
 to 86
 vfstab file, *See* /etc/vfstab file
 vgrind command, 234
 vgrindefs database, 337
 vhangup system call, 258
 vi editor, 234, 341
 vidattr library routine, 323
 vidputs library routine, 323
 view command, 234
 vipw command, 234
 virtual file-system architecture (VFS), 82
 to 86
 vlimit library routine, 323
 vmstat command, 234
 /vmsys directory, 344
 /vmunix file, 17, 25, 71, 178, 180
 /vol file system, 79, 80, 81
 /vol/dev/aliases/cdrom0 file
 system, 67
 /vol/dev/aliases/floppy0 file
 system, 67
 volcancel command, 67
 volcheck command, 68
 volcopy command, 85, 101
 vold command, 68, 79
 vold.conf file, 68
 volmissing command, 68
Volume Management
 commands, 67
 configuration files, 68
 default file system for, 79
 error messages, 67
 overview, 22, 66
 using, 66 to 68
 vplot command, 234
 vprintf library routine, 324

`vsprintf` library routine, 324
`vswap` command, 234
`vsyslog` library routine, 324
`vtimes` library routine, 324
`vtroff` command, 234
`vwidth` command, 234
`vwprintw` library routine, 324
`wscanw` library routine, 324

W

`w` command, 234
`waddch` library routine, 324
`waddstr` library routine, 325
`wait` command, 234
`wait` system call, 258, 259
`wait3` system call, 258
`wait4` system call, 259
`waitpid` system call, 258, 259
`wakeup` command, 190
`wall` command, 234
`wattroff` library routine, 325
`wattron` library routine, 325
`wattrset` library routine, 325
`wc` command, 234
`wclear` library routine, 325
`wclrtoeb` library routine, 325
`wclrtoeol` library routine, 326
`wcstombs` library routine, 326
`wctomb` library routine, 326
`wdelch` library routine, 326
`wdeleteln` library routine, 326
`wechochar` library routine, 326
`werase` library routine, 327
`WEXITSTATUS` system call, 239
`wgetch` library routine, 327
`wgetstr` library routine, 327
`what` command, 169, 234
`whatis` command, 234
`whatis` database, 58, 199
`whereis` command, 234

`which` command, 234
`who` command, 233, 234
`whoami` command, 234
`whois` command, 235
`WIFEXITED` system call, 239
`WIFSIGNALED` system call, 239
`WIFSTOPPED` system call, 239
`WIFSTOPSIG` system call, 239
`winch` library routine, 327
`windex` database, 58, 199
`window` management services, 6
`window` system, default, 52
`winsch` library routine, 327
`winsertln` library routine, 328
`wmove` library routine, 328
`wnoutrefresh` library routine, 328
`workspace` manager, 6
`wprintw` library routine, 328
`wrefresh` library routine, 329
`write` command, 235
`write` system call, 260
`writev` system call, 261
`wscanw` library routine, 329
`wsetscrreg` library routine, 329
`wstandend` library routine, 329
`wstandout` library routine, 330
`WTERMSIG` system call, 239

X

`xargs` command, 235
`xcrypt` library routine, 330
`xdecrypt` library routine, 330
`xdr_accepted_reply` library routine, 330
`xdr_array` library routine, 330
`xdr_authsys_parms` library routine, 330
`xdr_authunix_parms` library routine, 330
`xdr_bool` library routine, 330

xdr_bytes library routine, 330
xdr_callhdr library routine, 330
xdr_callmsg library routine, 330
xdr_char library routine, 330
xdr_destroy library routine, 330
xdr_double library routine, 330
xdr_enum library routine, 330
xdr_float library routine, 330
xdr_free library routine, 330
xdr_getpos library routine, 330
xdr_inline library routine, 331
xdr_int library routine, 331
xdr_long library routine, 331
xdr_opaque library routine, 331
xdr_pointer library routine, 331
xdr_reference library routine, 331
xdr_setpos library routine, 331
xdr_short library routine, 331
xdr_string library routine, 331
xdr_u_char library routine, 331
xdr_u_int library routine, 331
xdr_u_long library routine, 331
xdr_u_short library routine, 331
xdr_union library routine, 331
xdr_vector library routine, 331
xdr_void library routine, 331
xdr_wrapstring library routine, 331
xdrmem_create library routine, 331
xdrrec_create library routine, 331
xdrrec_endofrecord library routine, 331
xdrrec_eof library routine, 331
xdrrec_skiprecord library routine, 331
xdrstdio_create library routine, 331
XENIX semaphore file system (xnamefs), 84
XENIX, Solaris SPARC release and, 9
xget command, 235
xgettext command, 174
xlock command (OpenWindows), 215
xnamefs, 84
 /xpg2bin directory, 93
 /xpg2include directory, 93
 /xpg2lib directory, *See* /usr/xpg2lib directory
xsend command, 235
xset command (OpenWindows), 226
xstr command, 235
xtab system file, 337, 347
xtom library routine, 331
xv_get_sel command, 210
XView File Chooser, OpenWindows Developer's Guide File Chooser vs., 52
XView Window Toolkit, 167

Y

y0 library routine, 332
y1 library routine, 332
yacc command, 169, 235
yaccpar command, 169
yes command, 235
yn library routine, 332
yp database directory, 342
yp prefix, 19
ypaliases system file, 337
ypbatchupd command, 235
ypbind command, 235, 347
ypcat command, 235
ypfiles database, 337
ypgroup system file, 337
ypinit command, 235
ypmatch command, 235
yppasswd command, 235
yppasswd system file, 337
yppoll command, 235
ypprintcap database, 337
yppush command, 235
ypserv command, 235

ypset command, 235
ypupdated command, 235
ypwhich command, 235
ypxfr command, 235
ypxfrd command, 235

Z

zcat command, 236
zdump command, 236
zic command, 236
/zoneinfo directory, 344

Copyright 1995 Sun Microsystems Inc., 2550 Garcia Avenue, Mountain View, Californie 94043-1100 U.S.A.

Tous droits réservés. Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, et la décompilation. Aucune partie de ce produit ou de sa documentation associée ne peuvent être reproduits sous aucune forme, par quelque moyen que ce soit sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il en a.

Des parties de ce produit pourront être dérivées du système UNIX[®], licencié par UNIX System Laboratories, Inc., filiale entièrement détenue par Novell, Inc., ainsi que par le système 4.3. de Berkeley, licencié par l'Université de Californie. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

LEGENDE RELATIVE AUX DROITS RESTREINTS: l'utilisation, la duplication ou la divulgation par l'administration américaine sont soumises aux restrictions visées à l'alinéa (c)(1)(ii) de la clause relative aux droits des données techniques et aux logiciels informatiques du DFARS 252.227-7013 et FAR 52.227-19. Le produit décrit dans ce manuel peut être protégé par un ou plusieurs brevet(s) américain(s), étranger(s) ou par des demandes en cours d'enregistrement.

MARQUES

Sun, Sun Microsystems, le logo Sun, SunSoft, le logo SunSoft, Solaris, SunOS, OpenWindows, DeskSet, ONC, ONC+ et NFS sont des marques déposées ou enregistrées par Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. UNIX est une marque enregistrée aux Etats-Unis et dans d'autres pays, et exclusivement licenciée par X/Open Company Ltd. OPEN LOOK est une marque enregistrée de Novell, Inc. PostScript et Display PostScript sont des marques d'Adobe Systems, Inc.

Toutes les marques SPARC sont des marques déposées ou enregistrées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. SPARCcenter, SPARCcluster, SPARCcompiler, SPARCdesign, SPARC811, SPARCengine, SPARCprinter, SPARCserver, SPARCstation, SPARCstorage, SPARCworks, microSPARC, microSPARC-II, et UltraSPARC sont exclusivement licenciées à Sun Microsystems, Inc. Les produits portant les marques sont basés sur une architecture développée par Sun Microsystems, Inc.

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

Le système X Window est un produit du X Consortium, Inc.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" SANS GARANTIE D'AUCUNE SORTE, NI EXPRESSE NI IMPLICITE, Y COMPRIS, ET SANS QUE CETTE LISTE NE SOIT LIMITATIVE, DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DES PRODUITS A REpondre A UNE UTILISATION PARTICULIERE OU LE FAIT QU'ILS NE SOIENT PAS CONTREFAISANTS DE PRODUITS DE TIERS.

CETTE PUBLICATION PEUT CONTENIR DES MENTIONS TECHNIQUES ERRONEES OU DES ERREURS TYPOGRAPHIQUES. DES CHANGEMENTS SONT PERIODIQUEMENT APPORTES AUX INFORMATIONS CONTENUES AUX PRESENTES. CES CHANGEMENTS SERONT INCORPORES AUX NOUVELLES EDITIONS DE LA PUBLICATION. SUN MICROSYSTEMS INC. PEUT REALISER DES AMELIORATIONS ET/OU DES CHANGEMENTS DANS LE(S) PRODUIT(S) ET/OU LE(S) PROGRAMME(S) DECRITS DANS CETTE PUBLICATION A TOUS MOMENTS.

