

Solaris 1.x to Solaris 2.x Transition Guide

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



© 1994 Sun Microsystems, Inc.
2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A.

All rights reserved. This product and related documentation are protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or related documentation 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[®] and Berkeley 4.3 BSD systems, licensed from UNIX System Laboratories, Inc., a wholly owned subsidiary of Novell, Inc., and the University of California, respectively. Third-party font software in this product is protected by copyright and licensed from Sun's font suppliers.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the United States Government is subject to the restrictions set forth in DFARS 252.227-7013 (c)(1)(ii) 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, the Sun logo, Sun Microsystems, Sun Microsystems Computer Corporation, SunSoft, the SunSoft logo, Solaris, SunOS, OpenWindows, DeskSet, ONC, ONC+, and NFS are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and certain other countries. UNIX is a registered trademark of Novell, Inc., in the United States and other countries; X/Open Company, Ltd., is the exclusive licensor of such trademark. OPEN LOOK[®] is a registered trademark of Novell, Inc. PostScript and Display PostScript are trademarks of Adobe Systems, Inc. All other product names mentioned herein are the trademarks of their respective owners.

All SPARC trademarks, including the SCD Compliant Logo, are trademarks or registered trademarks of SPARC International, Inc. SPARCstation, SPARCserver, SPARCengine, SPARCstorage, SPARCware, SPARCcenter, SPARCclassic, SPARCcluster, SPARCdesign, SPARC811, SPARCprinter, UltraSPARC, microSPARC, SPARCworks, and SPARCcompiler 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 product of the Massachusetts Institute of Technology.

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 PROGRAM(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
Administration Tool	18
Network Information Service Plus (NIS+)	21
Print Subsystem	21
Printer Manager.	22
PrintTool and Command Changes	22
Service Access Facility.	22
Volume Management.	24
3. Installation and Configuration Changes.	25
Installation Overview	25
What's New in Installation and Configuration	26
What to Do Before You Install Solaris Software.	27
Saving Disk Partition Information.	28
Making a List of System Components to Save.	29
Saving File System Information.	29
Making a List of Files and File Systems to Save	30
Saving Metadevice Configuration Information.	30
Deciding the Order of Installation for Networks.	31

Backing Up the File Systems	32
Saving SunOS Release 4.x System Data	32
Installing Solaris Software	33
Restoring Data	34
Restoring SunOS Release 4.x User Data	34
Restoring SunOS Release 4.x System Data.....	36
4. Using the Compatibility Packages.....	41
Why Port Applications?	42
SunOS/BSD Source Compatibility Package.....	42
Binary Compatibility Package	42
5. Security.....	45
Solaris 2.4 Security Features	46
/etc/passwd and /etc/shadow Files	46
/etc/default Files	47
Restricted Shells.....	47
Administration Tool Security.....	47
Password Aging Changes.....	48
Automated Security Enhancement Tool (ASET)	48
Unbundled Security Options	49
Kerberos Security.....	49
SunShield	50
6. User Environment Administration	51
Selecting a Default Shell	51
Customizing User Environments.....	53

Window System	54
User and Group Administration	55
User and Group Administration Choices	55
Adding User Accounts	55
The Administration Tool's User Account Manager	56
Using Mail	58
Using Document Tools	59
man Page Organization Differences	60
Customizing the man Command Search Path	61
whatis and windex Databases	62
Using the man Command	63
7. Device Administration	65
Device Naming Conventions	65
Convention for Disks	66
Convention for Tape Drives	67
Obtaining Disk Information	67
df Command	68
du Command	68
dtkinfo Command	68
devinfo Command	69
Adding Devices to the System	69
Using Volume Management	70
8. Startup and Shutdown	73
Booting	73

boot Command Changes.....	74
Booting From the PROM.....	74
Summary of Boot Differences.....	75
Using the <code>init</code> Command.....	76
init Command Changes.....	76
Changing System Run Levels.....	76
Shutting Down.....	78
Changes to the <code>shutdown</code> Command.....	79
Using the <code>fasthalt</code> and <code>fastboot</code> Commands.....	79
Using the <code>halt</code> and <code>reboot</code> Commands.....	80
9. File-System Administration	81
File-System Changes	82
Default File Systems and Directories.....	83
Virtual File-System Architecture	86
Supported File-System Types.....	86
Unsupported SVR4 File System Types	88
Generic File-System Commands	88
Directory and File Changes	91
<code>/dev</code> Directory.....	91
<code>/etc</code> Directory.....	91
<code>/sbin</code> Directory	96
<code>/usr</code> Directory.....	97
<code>/var</code> Directory.....	97
<code>/kernel</code> Directory	98

/opt Directory	98
/sys Directory	99
Using File System Administration Commands.....	99
Mounting File Systems and <code>autofs</code>	99
Monitoring File Systems.....	102
Sharing File Systems	103
Creating New File Systems.....	104
Checking File Systems.....	104
Backing Up and Restoring Files.....	105
10. Setting Up a Solaris 2.4 Server to Support SunOS Release 4.x Diskless/Dataless Clients	111
Adding SunOS Release 4.x Support to a Solaris 2.4 Server ...	111
Running <code>discover4x</code>	113
Setting Up the CD-ROM Drive for <code>install4x</code>	114
Running <code>install4x</code>	115
Running <code>convert4x</code>	119
Using Host Manager to Add SunOS Release 4.x Diskless/Dataless Clients	119
11. Setting Up and Using Printers	121
Summary of Printing Differences	121
The Administration Tool's Printer Manager.....	122
Setting Up a Printer.....	124
Print Commands and the Compatibility Package	124
Using Printer Commands.....	125
Using SunOS 5.4 Printer Administration Commands.....	125

Printing troff.....	126
12. Managing Terminals and Modems	129
Terminal and Modem Management	129
Service Access Facility (SAF)	130
What's a Port Monitor?.....	130
sacadm Command	131
pmadm Command	131
SAF Functions and Related Programs.....	132
13. Network Service Administration.....	133
Changes to TCP/IP	133
Changes to NFS	134
PPP.....	134
UUCP.....	134
Checkpoint Restart	136
User Job Grades.....	136
Limits File	136
Config File	137
Log Files	137
14. Using Name Services	139
Name Service Switch.....	140
NIS+.....	140
DNS	140
DNS and NIS+ Comparison	141
NIS and NIS+ Comparison.....	142

NIS Support in SunOS Release 5.4.	143
Planning an NIS+ Upgrade.	143
Installing NIS+ on a SunOS Release 4.x System.	143
<i>Part 2—Transition Information for Developers</i>	
15. Compilers, Linkers, and Debuggers	147
Compilers	147
Compiler Option Differences	148
Linkers	154
Link Editor Option Differences	154
Building Shared Libraries.	156
Building Executables.	156
Specifying Library Search Paths	157
Search Path Rules.	158
Version Numbering.	158
Examples.	159
Debuggers.	162
dbx and dbxtool	162
adb and kadb	162
kadb Macros.	162
Debugging a Live Kernel	163
truss Command.	163
16. Tools and Resources	165
ioctl() Requests.	165
ptrace() Request Values	168

Libraries	170
Shared Libraries.....	170
Resource Limits	170
Using make	172
Using SCCS.....	173
Determining Application Compatibility.....	173
Well-Behaved Applications	174
Dynamically Linked and Statically Linked Executables. . .	174
Packaging Applications	174
Packaging Utilities.....	175
Toolkits	176
OLIT	176
XView	176
Finding SunOS Release 4.x Tools	176
17. Networking and Internationalization.....	181
Networking.....	181
NIS, NIS+	182
nsswitch.conf File	182
NIT.....	182
Sockets.....	182
Internationalization.....	183
Character Support.....	183
Message Catalogs.....	184
Commands	184

Libraries	184
18. System and Device Configuration	187
System Configuration	187
Dynamically Loaded Kernel	187
Kernel Layout	188
config Command	188
/etc/system File	189
boot Command	189
Summary of Boot Differences	190
Device Configuration: The Reconfiguration Boot	191
Device Naming From a Developer's Perspective	191
/devices	191
/dev	192
Device Driver Naming	192
19. Device Drivers and STREAMS	195
Device Drivers and STREAMS Device Drivers	195
Device Driver Interfaces	196
devinfo Command	198
Porting Considerations	199
STREAMS	201
Solaris 2.x Driver Architecture	202
Device Driver Commands	203
A. Commands Reference Table	205
Using the Reference Table	205

Examples	206
The Commands Reference Table	207
B. System Calls Reference Table	249
Using the Reference Table	249
Examples	250
System Calls	251
C. Library Routines Reference Table	275
Using the Reference Table	275
Examples	276
Library Routines	278
D. System Files Reference Table	347
Using the Reference Table	347
System Files	348
E. / and /usr File Systems Changes	353
Layout of the / File System	353
Layout of the /usr File System	356
F. Quick Reference for Basic Changes	359

Figures

Figure 2-1	Disk Slices and Cylinder Groups	14
Figure 2-2	Administration Tool's GUI	19
Figure 7-1	SunOS release 5.4 <code>prtvtoc</code> Command Output.....	69
Figure 9-1	Solaris Release 2.4 Default File Systems and Directory Hierarchy	84
Figure 9-2	SunOS Release 5.4 <code>/etc/vfstab</code> file	96
Figure 11-1	Printer Manager Functions.....	122

Tables

Table 2-1	SAF Functions and Associated Programs	24
Table 5-1	Files in /etc/default Directory	47
Table 6-1	Basic Features of the Bourne, C, and Korn Shells.	52
Table 6-2	Initialization Files for Bourne, C, and Korn Shells	53
Table 6-3	Default Home Directory Startup Files	53
Table 6-4	Administration Tool Requirements	57
Table 6-5	SunOS Release 5.4 man Page Directories	60
Table 6-6	New man Command Options	63
Table 7-1	SunOS Release 4.x and SunOS Release 5.4 Device Names	67
Table 7-2	Location of CD-ROM and a Diskette With a File System	70
Table 7-3	Location of a CD-ROM and a Diskette Without a File System	71
Table 7-4	Volume Management Commands	72
Table 8-1	SunOS Release 5.4 Replacements for <code>reboot</code>	74
Table 8-2	Summary of Booting Differences	75
Table 8-3	SunOS Release 5.4 System Initialization Run Levels	77
Table 8-4	SunOS Release 5.4 Replacements for <code>shutdown</code>	78

Table 8-5	SunOS Release 5.4 shutdown Command Options	79
Table 9-1	Default Solaris 2.4 File Systems and Directories	85
Table 9-2	Summary of File System Types	86
Table 9-3	SVR4 File System Types Not Supported	88
Table 9-4	Generic File System Administrative Commands	88
Table 9-5	Locations of File System Commands	90
Table 9-6	Additions to the /dev Directory	91
Table 9-7	Initialization Scripts and Their Run Control Files	92
Table 9-8	Additions to the /etc Directory	92
Table 9-9	/etc/vfstab File Field Names and Content	94
Table 9-10	Additions to the /usr Directory	97
Table 9-11	Files Changed in the /usr Directory	97
Table 9-12	Additions to the /kernel Directory	98
Table 9-13	SunOS Release 4.x and SunOS Release 5.4 mount Command Differences	100
Table 9-14	File and Directory Monitoring Commands	102
Table 9-15	ufsdump Command Options Not Available With the dump Command	106
Table 9-16	Tape Drive Density Characters in Tape Device Names	108
Table 9-17	cpio Options Added Since SunOS Release 4.x	108
Table 10-1	Information Needed to Add Network Clients	120
Table 11-1	Information Needed to Set up a Local Printer	123
Table 11-2	Information Needed to Enable Access to a Remote Printer	124
Table 11-3	User Print Command Equivalents	125
Table 11-4	Printer Administration, Setup, and File Equivalents	126
Table 11-5	SunOS Release 5.4 troff commands	127

Table 12-1	SAF Functions and Associated Programs	132
Table 13-1	New SunOS Release 5.4 UUCP Files and Commands	135
Table 13-2	New SunOS Release 5.4 UUCP Log Files.	137
Table 14-1	DNS and NIS+ Features and Benefits Compared.	141
Table 14-2	NIS and NIS+ Features Compared	142
Table 15-1	Comparison of C Compiler Options.	148
Table 15-2	Filename Extensions Used by the C Compiler	153
Table 15-3	Comparison of ld Options.	154
Table 15-4	SunOS Release 4.x Linker Search Paths.	158
Table 15-5	Solaris 2.4 Linker Search Paths With LD_LIBRARY_PATH= <i>dirlist1</i> 158	
Table 15-6	Solaris 2.4 Linker Search Paths With LD_LIBRARY_PATH= <i>dirlist1,dirlist2</i>	158
Table 15-7	Example Shared Libraries	159
Table 16-1	Unsupported ioctl() Requests	166
Table 16-2	Unsupported ttycom ioctl Requests.	166
Table 16-3	ioctl() Support	167
Table 16-4	ptrace() Requests not Supported by Solaris 2.4 Software . .	169
Table 16-5	Resource Limits	170
Table 16-6	Comparison of Library Locations	171
Table 16-7	Commands for Creating Packages	175
Table 16-8	Commands for Adding and Removing Packages	175
Table 16-9	Commands for Providing Information About Packages.	175
Table 16-10	Bundled Programming Tools	176
Table 16-11	New Programming Tools	178
Table 16-12	Unbundled Programming Tools	178

Table 17-1	xpg2lib Library Routine Locations	185
Table 18-1	Summary of Boot Differences	190
Table A-1	Example 1	206
Table A-2	Example 2	206
Table A-3	Example 3	207
Table A-4	Example 4	207
Table A-5	Commands Reference Table	208
Table B-1	System Calls Reference Table	251
Table C-1	Library Routines Reference Table	278
Table D-1	File Formats Reference Table	348
Table E-1	Directories in the / File System	353
Table E-2	Directories in the /usr File System	356
Table F-1	Basic Commands	359
Table F-2	Advanced Commands	360
Table F-3	Files and Directories	361
Table F-4	Daemons and Standard Processes	361

Preface

The Solaris™ 2.4 operating environment, SunSoft's™ distributed computing solution, is comprised of SunOS™ release 5.4 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 5.4 operating systems for people already familiar with SunOS release 4.x. This guide also handles other aspects of the Solaris 2.4 operating environment that can help you through the transition.

If you are looking for more information about features now available with the Solaris 2.4 operating environment, see *Solaris 2.4 Introduction*.

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 2.4 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 5.4 operating environments to make your transition to the Solaris 2.4 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 2.4 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 2.4 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 5.4. 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 2.4 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 2.4 operating environment.

-
- **Chapter 5, “Security,”** describes the major differences between SunOS release 4.x and Solaris 2.4 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 5.4 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 2.4 Server to Support SunOS Release 4.x Diskless/Dataless Clients,”** discusses using the Administration Tool’s Host Manager to set up servers for clients and it describes three programs, `discover4x`, `install4x`, and `convert4x`, that work together to help prepare a Solaris 2.4 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 2.4 software. It also describes the changes to printer commands.
 - **Chapter 12, “Managing Terminals and Modems,”** describes terminal and modem management through the Administration Tool’s Serial Port Manager and it describes the Service Access Facility (SAF), a Solaris 2.4 tool for managing access to local and network system services, such as modems and terminals.
 - **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 2.4 packaging capabilities, and how to find SunOS release 4.x tools.
- **Chapter 17, “Networking and Internationalization,”** discusses Solaris 2.4 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 2.4 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 5.4 commands.
- **Appendix B, “System Calls Reference Table,”** compares SunOS release 4.x and SunOS release 5.4 system calls.

- **Appendix C, “Library Routines Reference Table,”** compares SunOS release 4.x and SunOS release 5.4 library routines.
- **Appendix D, “System Files Reference Table,”** compares SunOS release 4.x and SunOS release 5.4 system files.
- **Appendix E, “/ and /usr File Systems Changes,”** compares SunOS release 4.x and SunOS release 5.4 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 2.4 operating environment, see the following documentation:

- *Solaris 2.4 Introduction*
- *Solaris User's Guide*
- *Solaris Advanced User's Guide*
- *SPARC: Installing Solaris Software*
- *User Accounts, Printers, and Mail Administration*
- *Common Administration Tasks*
- *File System Administration*
- *Peripherals Administration*
- *Name Services Administration Guide*
- *Name Services Configuration Guide*
- *NIS+ Transition Guide*
- *Security, Performance, and Accounting Administration*
- *TCP/IP Network Administration Guide*
- *man Pages(1M): System Administration Commands* or on line in *SunOS Reference Manual*
- *Solaris Binary Compatibility Guide*
- *Solaris 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 2.4 software, and to understand changes to the local computing environment, and to understand changes to routine tasks.

Introduction

1 

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 5.4* is a full 32-bit operating system based on UNIX System V Release 4 (SVR4)
- *Cross-functional compatibility* enables SunOS release 5.4 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 Administration Tool* is the base for object-oriented 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.
- *Administration Tool.* The Administration Tool, which runs under the OpenWindows environment, provides system management facilities to help add hosts, manage the network, and many other routine tasks.
- *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 to 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 2.4 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>Administration Tool</i>	<i>page 18</i>
<i>Network Information Service Plus (NIS+)</i>	<i>page 21</i>
<i>Print Subsystem</i>	<i>page 21</i>
<i>Service Access Facility</i>	<i>page 22</i>
<i>Volume Management</i>	<i>page 24</i>

Software Packages and Clusters

Solaris 2.4 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 2.4 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 2.4 operating environment. Compilers and debuggers are not included in the Solaris 2.4 operating environment.
- *Entire Distribution* contains the entire Solaris 2.4 environment.

The following subsections briefly discuss package administration. For more information about this section's topics, see *Common Administration Tasks*.

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 `swmtool(1)` man page)
- The command-line utilities (see the `pkgadd(1M)` and `pkgrm(1M)` man pages)

Graphical User Interface (`swmtool`)

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

Use the Software Manager to:

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

If you want to install or remove the software from a local or remote system, you must run Software Manager as superuser. 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 *Peripherals Administration* 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 the *Administration Supplement for Solaris Platforms* 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 *Peripherals Administration* 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.

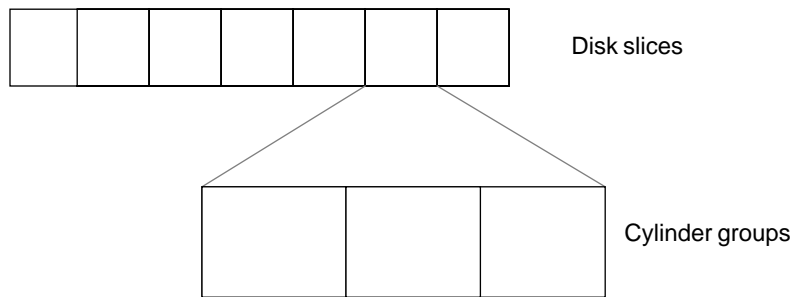


Figure 2-1 Disk Slices and Cylinder Groups

Device Naming

SunOS release 5.4 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 5.4 conventions are slightly different from AT&T SVR4 device names because SunOS release 5.4 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 91). For discussions on device-naming conventions, see “Device Naming Conventions” on page 65 and “Device Naming From a Developer’s Perspective” on page 191.

File Systems

SunOS release 5.4 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. *File System Administration* 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 5.4.
- The `/etc/vfstab` tab file replaces `/etc/fstab`.
- The `/etc/lp` directory replaces `/etc/printcap`.
- The SunOS release 5.4 `/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 5.4 `/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 5.4.
- 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 5.4, 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 5.4.

The pseudo file systems in SunOS release 5.4 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 5.4 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 5.4 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 99 for more detailed information.

Administration Tool

One of the major changes affecting system administration between SunOS release 4.x and SunOS release 5.x is the availability of the Administration Tool with OpenWindows. This powerful 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 and remote systems if the right access has been set up.

Administration Tool applications enable you to manage the following:

- Important 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
- Diskless and dataless client setup

Using a graphical user interface (GUI) like the Administration Tool 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

The Administration Tool's GUI is shown in Figure 2-2.

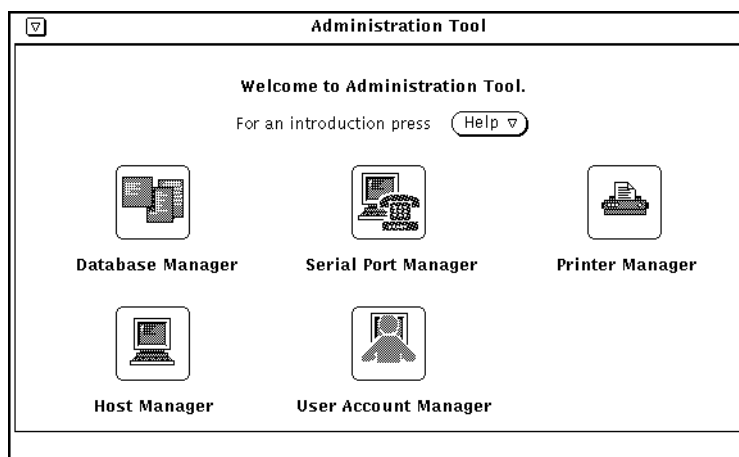


Figure 2-2 Administration Tool's GUI

As the figure shows, the Administration Tool's main features represent five areas of system administration and may be accessed through icons for the following:

- *Database Manager* to update NIS+ maps or `/etc` files, or to browse NIS maps
- *Printer Manager* to add, remove, and maintain local and remote printers
- *User Manager* to add and maintain user accounts and groups
- *Host Manager* to define clients on a server
- *Serial Port Manager* to manage terminals and modems

Note – You do not need to be root to start the Administration Tool, 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 the Administration Tool window shown in Figure 2-2, type the following command in any window.

```
$ admintool &
```

This command displays the window shown in Figure 2-2.

If you are using NIS+ or you are not using any naming service, you can use the Administration Tool to create and maintain user accounts and groups.

Note – You can use the Administration Tool to browse information in NIS maps, but you cannot modify NIS maps with it.

If you are not using a naming service, you have to repeat changes to the `/etc` files for each system. To change the `/etc` files on any system other than the local system, specify the host name when you select the naming service “None” in the Administration Tool's Load Database window (if you have permission on the remote system).

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*, *Name Services Administration Guide*, and *Name Services Configuration Guide* for more information.

Print Subsystem

The print management commands have changed between SunOS release 4.x and the Solaris 2.4 operating environment. In the Solaris 2.4 operating environment, you can use command-line procedures or the Administration Tool 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 *User Accounts, Printers, and Mail Administration* for more information.

Printer Manager

Printer Manager is an OPEN LOOK printer server and print client installation configuration application for the bundled SVR4 LP print subsystem. Printer Manager simplifies printer system administration and easily allows connections to SunOS release 4.x print servers.

By registering the printer server and printer name with NIS+, any Solaris 2.4 client can browse the available printers from a list, eliminating the need to know a printer name and server name in advance.

PrintTool and Command Changes

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 2.4 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 2.4

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

3 

Migrating from SunOS release 4.x to the Solaris 2.4 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 2.4 Server to Support SunOS Release 4.x Diskless/Dataless Clients,” for information about creating an environment that serves both Solaris 2.4 and SunOS release 4.x clients.)

This chapter contains the following sections:

<i>Installation Overview</i>	<i>page 25</i>
<i>What to Do Before You Install Solaris Software</i>	<i>page 27</i>
<i>Installing Solaris Software</i>	<i>page 33</i>
<i>Restoring Data</i>	<i>page 34</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 5.4 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 2.4 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 2.4 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 5.4 installation introduces a number of changes in the way administrators install software on existing systems and add new systems to the network.

- SunOS release 5.4 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 5.4 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 5.4 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 5.4 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 5.4 in a limited environment for testing. You can add or remove packages later as you gain more experience with the system.
- SunOS release 5.4 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`.
- Host Manager, an option under the Administration Tool, allows you to add information to the network databases about a new system.

What to Do Before You Install Solaris Software

Converting a SunOS release 4.x system to a Solaris 2.4 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 2.4 software. Because the device naming conventions are different in the Solaris 2.4 operating environment, it is possible

to become confused and inadvertently choose the wrong disk when you install the Solaris 2.4 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 `dkinfo(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/dkinfo 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 2.4 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 Online: 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 Online: 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 Online: 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 2.4 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 2.4 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 2.4 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 2.4 software. If the system will serve both Solaris 2.4 clients and SunOS release 4.x clients, also see Chapter 10, “Setting Up a Solaris 2.4 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 5.4 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 2.4. 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 65 if you need help with Solaris 2.4 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 5.4. Before running them, examine the files and replace any incompatible SunOS release 4.x commands with their SunOS release 5.4 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 5.4 work environment, select the same login shell in the SunOS release 5.4 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 65 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 5.4 `/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 2.4 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 2.4 environment. For example, `/etc/printcap` entries must be typed into the Solaris 2.4 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 2.4 software.

You must change the following files before merging the data onto the Solaris 2.4 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 5.4.
- `/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 the *User Accounts, Printers, and Mail Administration* for additional information.

Using the Compatibility Packages



SunOS release 5.4 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 2.4 operating environment. Compatibility packages make it possible for these programs to run on a Solaris 2.4 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 2.4 operating environment.

This chapter includes the following sections:

<i>Why Port Applications?</i>	<i>page 42</i>
<i>SunOS/BSD Source Compatibility Package</i>	<i>page 42</i>
<i>Binary Compatibility Package</i>	<i>page 42</i>

Some SunOS release 4.x commands are not available in the Solaris 2.4 operating environment. Others exist, but have changed. For information about changes to SunOS release 4.x commands in the Solaris 2.4 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 2.4 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 2.4 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 2.4 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 5.4 interfaces. These interfaces provide a familiar SunOS environment while your environment and applications are migrating to SunOS release 5.4. 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 *Solaris Source Compatibility Guide*.

Binary Compatibility Package

The Binary Compatibility Package is an optional package available with the Solaris 2.4 operating environment. The package allows existing SunOS release 4.x applications, both statically and dynamically linked, to run under the Solaris 2.4 operating environment without modification or recompilation. It

handles most binary interface discrepancies between the two releases transparently. This results in a Solaris 2.4 operating environment where SunOS release 4.x applications can run properly.

See *Solaris 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 2.4 operating environment, making them available for use before they are ported to SunOS release 5.4. With this package, well-behaved application binaries based on SunOS release 4.x system software will run under SunOS release 5.4 without modifications or recompilation.

The Binary Compatibility Package is intended for end-user environments, not for use as a development environment. All SunOS release 5.4 application development should be done under the base SunOS release 5.4 environment.

Security for the Solaris 2.4 operating environment combines several features from SunOS release 4.x and AT&T SVR4 with capabilities added specifically for Solaris 2.4. 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 2.4 operating environment security, and points out how those changes may affect system administration procedures. *Security, Performance, and Accounting Administration* describes the administration and use of these features more fully.

This chapter contains the following sections:

<i>Solaris 2.4 Security Features</i>	<i>page 46</i>
<i>Password Aging Changes</i>	<i>page 48</i>
<i>Automated Security Enhancement Tool (ASET)</i>	<i>page 48</i>
<i>Unbundled Security Options</i>	<i>page 49</i>

For more information about password aging and security in general, see *Security, Performance, and Accounting Administration*.

Solaris 2.4 Security Features

Most of the security features from SunOS release 4.x systems are also available in the Solaris 2.4 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 Network Administration Guide* describes administering Internet security.

Security for local SunOS release 5.4 systems includes storing encrypted passwords in a separate file, controlling login defaults, and restricted shells. Equivalent NIS+ security, described in *Name Services 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 5.4 `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 *User Accounts, Printers, and Mail Administration*.

`/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

Administration Tool Security

The security of the Administration Tool and its Host Manager and Database Manager introduce the concept of allowing nonroot users to administer systems. Any authorized user can use Administration Tool capabilities to display information the Administration Tool controls. However, the ability to

make system administrative changes depends on other security decisions. For example, you must plan the type of access NIS+ allows, taking into consideration Administration Tool capabilities. See *Name Services Administration Guide*.

Password Aging Changes

The SunOS release 5.4 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 the User Account Manager feature of the Administration Tool if you are running OpenWindows. For information about this method, see *Security, Performance, and Accounting Administration*.
- *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 the User Account Manager feature of the Administration Tool if you are running OpenWindows. For information about this method, see *Security, Performance, and Accounting Administration*.

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 2.4 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

Security, Performance, and Accounting Administration describes ASET setup and monitoring in detail.

Unbundled Security Options

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

Kerberos Security

The Solaris 2.4 operating environment includes support for Kerberos authentication for secure RPC. Kerberos source code and administrative utilities are available from MIT.

Solaris 2.4 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

Security, Performance, and Accounting Administration describes how to set up and administer SunOS release 5.4 Kerberos. *System Services 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 2.4 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.

This chapter describes differences in tasks you may perform to set up the local user environment after installing the Solaris 2.4 software.

<i>Selecting a Default Shell</i>	<i>page 51</i>
<i>Customizing User Environments</i>	<i>page 53</i>
<i>Window System</i>	<i>page 54</i>
<i>User and Group Administration</i>	<i>page 55</i>
<i>Using Mail</i>	<i>page 58</i>
<i>Using Document Tools</i>	<i>page 59</i>
<i>man Page Organization Differences</i>	<i>page 60</i>

Selecting a Default Shell

The login shell is the command interpreter that runs when you are logged in. The Solaris 2.4 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 the Administration Tool. See *User Accounts, Printers, and Mail Administration* 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 5.4 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 *User Accounts, Printers, and Mail Administration*.

Using the SunOS 4.x Work Environment With Solaris

If you prefer to continue using the SunOS release 4.x work environment, SunOS release 5.4 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 2.4 operating environment.

Window System

OpenWindows 3.2 is the default Solaris 2.4 windowing environment. The OpenWindows environment offers a simple and intuitive interface for users of Solaris 2.4 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 5.4 and its usage is correct.

SunView™ software is not part of the Solaris 2.4 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 User Account Manager, available in the Administration Tool, they do allow you to do most of the tasks supported by the User Account Manager 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 2.4 operating environment, there are three ways to add (and maintain) user accounts:

- Use the User Account Manager in the Administration Tool – 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 the Administration Tool.
- Manually edit files (similar to the SunOS release 4.x procedure with a few exceptions)

Note – Because SunOS release 5.4 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.

User Accounts, Printers, and Mail Administration describes in detail the policy decisions you should consider before you begin to set up accounts. It also discusses the process for adding and modifying accounts, using the Administration Tool and the SunOS release 5.4 commands. *Security, Performance, and Accounting Administration* explains security considerations for controlling user access to systems and networks.

The Administration Tool's User Account Manager

The Administration Tool's User Account Manager enables you to set up user accounts on a system or network by filling in a series of pop-up windows. You can set up accounts with NIS+ or without a naming service. You can run the Administration Tool from any system on the network to make changes to the central NIS+ databases or to the `/etc` files on any system, provided you are a member in the `sysadmin` group (GID 14).

Because the User Account Manager can automatically copy user initialization files into their appropriate location, you should set up these files before using the User Account Manager. See *User Accounts, Printers, and Mail Administration* if you need more information.

Before you invoke the Administration Tool, you should have the information shown in Table 6-4.

Table 6-4 Administration Tool Requirements

Item Name	Description
User Name	Unique name by which the user is known to the system; Valid characters are alphanumerics plus the underscore (_); no other characters are acceptable. First character must be a letter; name must include at least one lowercase letter.
User ID	Unique number in the range 100–60,000 that identifies the user to the system.
Primary Group	Group identifier associated with a particular group. Range is 100–60,000; default is 100.
Secondary Groups	One or more group names or numbers to which the user also belongs. If you type more than one name or number, separate them with commas. Default is none.
Comment	Information such as user's full name, phone number, and organization.
Login Shell	Name of program that starts at login: <code>/bin/sh</code> for the Bourne shell, <code>/bin/csh</code> for the C shell, or <code>/bin/ksh</code> for the Korn shell. Default is the Bourne shell.
Password	Password status setting.
Min Change	Minimum number of days between password changes. Used to prevent a user from changing a password and immediately changing it back to the original password. Default is no minimum.
Max Change	Number of days the password may remain unchanged. If the password has not been changed within this number of days, the user will not be able to log in to the account. The administrator must reactivate the account. Default is no expiration.

Table 6-4 Administration Tool Requirements (Continued)

Item Name	Description
Max Inactive	Number of days an account can go unused (no login) before it is automatically locked. Default is no limit.
Expiration Date	Date on which the user account expires. Default is no expiration date.
Warning	Number of days before a password expires that the user is warned. Default is no warning.
Path	Full path to the home directory for this user account. By convention, this is <code>/export/home/user-name</code> .
Server	Host name (the name you assigned to your system when you installed the system)
Skeleton Path	The path name (usually <code>/etc/skel</code>) that contains the initialization files to be supplied for this user. Example: <code>.login</code> .
Permissions Owner: Group: World:	Permissions to read, write, and execute files by owner, group, and world in the home directory
Mail Server	Name of system where user's mailbox is located
Cred. Table Setup	NIS+ security entry for the user

Using Mail

The SunOS release 4.x mail programs are different in the Solaris 2.4 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 2.4 operating environment's version of `mail`. Additionally, some useful mail facilities are included for compatibility.

In the Solaris 2.4 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 2.4 `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 2.4 mail reading program. It is an enhanced version of SunOS release 4.x `/usr/ucb/mail`. In the Solaris 2.4 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 2.4 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 2.4 mail programs, see *User Accounts, Printers, and Mail Administration*.

Using Document Tools

This section outlines the main differences in using document tools between SunOS release 4.x and the Solaris 2.4 operating environment.

- The Solaris 2.4 operating environment provides a set of PostScript[®] filters and device-independent fonts. However, most SunOS release 4.x TranScript[®] filters have SunOS release 5.4 equivalents, while a few less common ones do not. In SunOS release 5.4 systems, there is no T_EX filter, no `pscat` (C/A/T) filter, and no raster image filter.

- The Solaris 2.4 operating environment provides device-independent `troff`, with the following features: SunOS release 4.x `troff` input files work with Solaris 2.4 `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 6-5 shows SunOS release 5.4 man page directories.

Table 6-5 SunOS Release 5.4 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 6-5 SunOS Release 5.4 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 5.4 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 5.4, 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 6-6 shows that SunOS release 5.4's version of the `man` command has additional search options.

Table 6-6 New `man` Command Options

Option	Description
<code>-a</code>	Displays all man pages that match <i>file name</i> . The pages are displayed sequentially in the order they are found.
<code>-l</code>	Lists all man pages that match the <i>file name</i> . 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 5.4 `man` command.

This chapter explains SunOS release 5.4 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 65</i>
<i>Obtaining Disk Information</i>	<i>page 67</i>
<i>Adding Devices to the System</i>	<i>page 69</i>
<i>Using Volume Management</i>	<i>page 70</i>

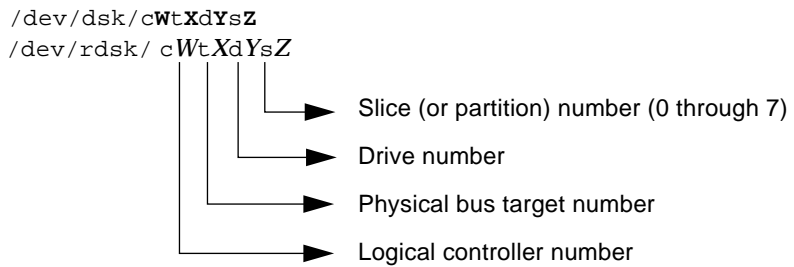
Device Naming Conventions

Device naming conventions have changed between SunOS release 4.x and SunOS release 5.4. 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 5.4 commands that take device names as arguments must use the SunOS release 5.4 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 *Solaris 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 5.4 `sd2a` is `c0t2d0s0` and `sd2b` is `c0t2d0s1`.

Convention for Tape Drives

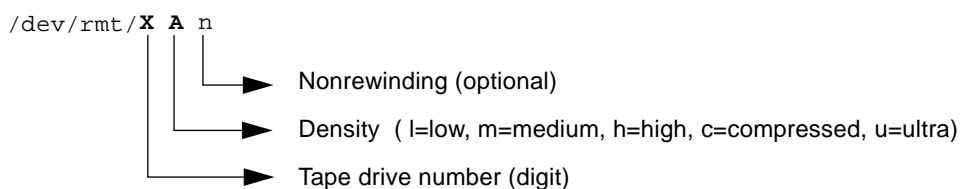


Table 7-1 provides some examples that compare the SunOS release 4.x and SunOS release 5.4 device naming conventions.

Table 7-1 SunOS Release 4.x and SunOS Release 5.4 Device Names

Device Description	SunOS release 4.x Device Name	SunOS release 5.4 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 5.4 release have changed. `df(1M)`, `du(1M)` are still available, but have changed. `dckinfo(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 5.4 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 5.4 version produces a full listing with totals.

Finally, use the SunOS release 5.4 device naming conventions when specifying special device names to this command. See “Device Naming Conventions” on page 65 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 *Peripherals Administration*.

Figure 7-1 shows output for the SunOS release 5.4 `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  Sector  Count  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 5.4 `prtvtoc` Command Output

`devinfo` Command

The SunOS release 4.x version of `devinfo` is incompatible with the SunOS release 5.4 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)`, see *File System Administration*. For more information about adding devices and drivers, see *Peripherals Administration*.

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/<i>cdrom_name</i></code>
Diskette	<code>/floppy/<i>floppy_name</i></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	<code>/vol/dev/aliases/cdrom0</code>
Diskette	<code>/vol/dev/aliases/floppy0</code>

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, see *Peripherals Administration*. For information on how to use and administer diskettes and CD-ROMs, see *File System Administration*.

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.

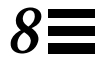
There are several new commands to help you administer Volume Management on your system as shown in Table 7-4.

Table 7-4 Volume Management Commands

Command	Description
rmmount(1)	Removable media mouter. Used by vold to automatically mount /cdrom and /floppy when media is installed.
volcancel(1)	Cancels a user's request to access a particular CD-ROM or diskette file system
volcheck(1)	Checks drive for installed media. By default, checks drive pointed to by /dev/diskette.
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 /etc/vold.conf

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 *Peripherals Administration* 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 73</i>
<i>Using the init Command</i>	<i>page 76</i>
<i>Shutting Down</i>	<i>page 78</i>

See “*Administration Supplement for Solaris Platforms*” 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 2.4 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 2.4 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 5.4.

In the Solaris 2.4 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 2.4 operating environment, but it brings the system down quickly, without orderly shutting down services. Table 8-1 shows the SunOS release 5.4 commands that replace those in SunOS release 4.x.

Table 8-1 SunOS Release 5.4 Replacements for `reboot`

SunOS release 4.x	SunOS release 5.4
<code>reboot</code>	<code>shutdown -i 6</code>
<code>fastboot</code>	<code>boot</code>

`boot` *Command Changes*

SunOS release 5.4 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 5.4 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 5.4 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 5.4	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 5.4	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 65.
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 5.4. 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 5.4 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 5.4 `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 5.4 `init` command lets you control the run level (initialization state) of your system and move easily between various modes of operation. SunOS release 5.4 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 5.4.

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

Table 8-3 SunOS Release 5.4 System Initialization Run Levels

Run Level	Default SunOS release 5.4 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 5.4 System Initialization Run Levels (Continued)

Run Level	Default SunOS release 5.4 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 5.4 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 “*Common Administration Tasks*” for detailed descriptions of shutdown procedures. `man` pages for each command are available in *man Pages(1): User Commands*.

In SunOS release 5.4, 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 5.4, but it stops the system quickly, without orderly shutting down services. Table 8-4 shows the SunOS release 5.4 commands that replace those in the SunOS release 4.x system.

Table 8-4 SunOS Release 5.4 Replacements for `shutdown`

SunOS release 4.x	SunOS release 5.4
<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 5.4 `shutdown` command includes only the options in Table 8-5. This command and its options are described in *File System Administration*.

Table 8-5 SunOS Release 5.4 `shutdown` Command Options

Option	SunOS release 5.4 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 77).
-y	Runs <code>shutdown</code> without asking confirmation questions. Assumes a “yes” response to all questions.

By default, the SunOS release 5.4 `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 5.4 `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 2.4 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 2.4 systems. The file-system checking features of these commands are not appropriate to Solaris 2.4.

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 5.4 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 82</i>
<i>Default File Systems and Directories</i>	<i>page 83</i>
<i>Virtual File-System Architecture</i>	<i>page 86</i>
<i>Directory and File Changes</i>	<i>page 91</i>
<i>Using File System Administration Commands</i>	<i>page 99</i>

For more information on understanding and managing file systems, see *File System Administration*.

File-System Changes

SunOS release 5.4 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 5.4.
- The `/etc/vfstab` tab file replaces `/etc/fstab`.
- The `/etc/lp` directory replaces `/etc/printcap`.
- The SunOS release 5.4 `/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 5.4 `/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 5.4.
- 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 5.4.

The added pseudo file systems in SunOS release 5.4 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 5.4 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 5.4 file system is hierarchical. Figure 9-1 graphically depicts SunOS release 5.4 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.

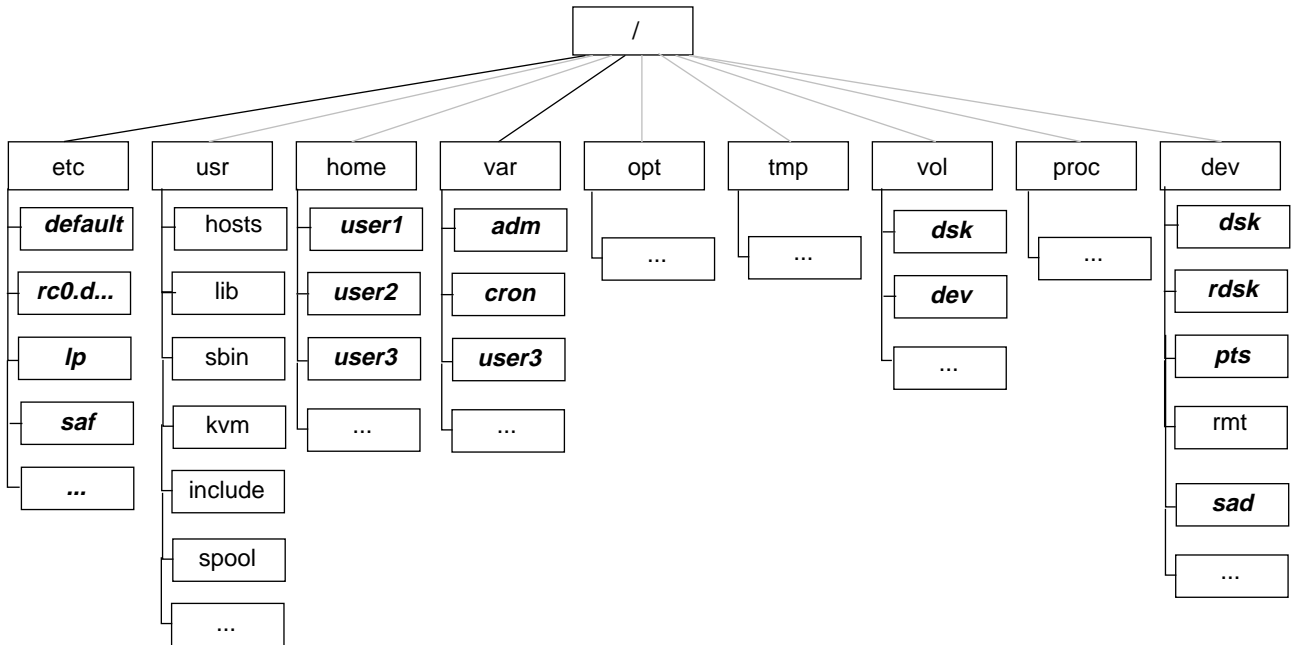


Figure 9-1 Solaris Release 2.4 Default File Systems and Directory Hierarchy

Solaris 2.4 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 5.4 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 5.4 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 5.4 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 5.4. There is one exception: The translucent file system (TFS) type has been withdrawn from SunOS release 5.4. Table 9-2 summarizes file-system type availability in SunOS release 4.x and SunOS release 5.4.

Table 9-2 Summary of File System Types

Category	Name	Description	SunOS Release 4.x	SunOS Release 5.4
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 5.4
	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 *File System Administration*.

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 5.4'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 5.4 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 5.4.

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 5.4, 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 5.4.

/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 5.4.

- 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 5.4. 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 5.4.

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 5.4, 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 88.

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/vfstab</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 5.4 `/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 5.4 `/etc/vfstab` file

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

The /etc/shadow File

The `/etc/shadow` file, new to SunOS release 5.4, 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 5.4 `/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 76 for a description of the scripts.

`/usr` Directory

The SunOS release 5.4 `/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 5.4.

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 5.4.

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

SunOS Release 4.x Location	SunOS Release 5.4 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 5.4.

- 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 5.4.

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

`/kernel` Directory

The SunOS release 5.4 `/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 5.4.

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 5.4 `/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 5.4 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 5.4, see *File System Administration* 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 *File System Administration*.

Changes to the `mount` Command

Some of the names and forms of the SunOS release 5.4 `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 5.4 `mount` Command Differences

SunOS Release 4.x	SunOS Release 5.4
<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 5.4 mount Command Differences

SunOS Release 4.x	SunOS Release 5.4
<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 70 for more information.

Specifying File Systems in the /etc/vfstab File

In the SunOS release 5.4 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 93.

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 5.4 <code>df</code> command. The SunOS release 5.4 <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 5.4 <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 5.4 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 5.4, 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 5.4 `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 5.4 device naming conventions (see “Device Naming Conventions” on page 65).

The mkfs Command

The SunOS release 5.4 `mkfs` command differs significantly from the SunOS release 4.x version of the command. The SunOS release 5.4 version provides for different file-system types, and its command syntax is entirely different (see “Generic File-System Commands” on page 88). Like `newfs`, `mkfs` accepts only names conforming to the SunOS release 5.4 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 5.4 `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 88).

In addition, `fsck` accepts only names conforming to the SunOS release 5.4 device naming conventions. For more information about the SunOS release 5.4 device naming conventions, see “Device Naming Conventions” on page 65.

The `fsck` command performs faster consistency checks at mount time. In addition, the SunOS release 5.4 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 5.4 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 5.4 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 5.4 system with `ufsrestore`.

SunOS release 5.4 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 5.4. 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 5.4 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 5.4 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 5.4 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 5.4 version, `w` denotes a unit of 2 bytes. In addition, the SunOS release 5.4 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 5.4, 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 5.4 `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 2.4 Server to Support SunOS Release 4.x Diskless/Dataless Clients

10 

This chapter outlines how to set up a Solaris 2.4 system as a server for SunOS release 4.x diskless/dataless clients. It discusses the two main steps necessary to set up servers for clients:

- Adding SunOS release 4.x support to a Solaris 2.4 server using the `discover4x`, `install4x`, and `convert4x` programs.
- Using Host Manager to add clients.

This chapter contains the following sections:

<i>Adding SunOS Release 4.x Support to a Solaris 2.4 Server</i>	<i>page 111</i>
<i>Using Host Manager to Add SunOS Release 4.x Diskless/Dataless Clients</i>	<i>page 119</i>

Make sure you have read Chapter 3, “Installation and Configuration Changes,” if you are setting up a Solaris 2.4 server for SunOS release 4.x clients on a Solaris 2.4 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 2.4 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 2.4 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 2.4 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 2.4 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 2.4 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 2.4 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 2.4 operating environment.
- `convert4x` – This program updates the Solaris 2.4 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 2.4 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 loading packages, see the *Software and AnswerBook Packages Administration Guide* section on adding and removing packages.

Running discover4x

`discover4x` analyzes the support that remains for SunOS release 4.x clients after the server has migrated to the Solaris 2.4 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 2.4 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 2.4 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 2.4 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 2.4 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 hfsfs -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 Host Manager. Use the following section to add *new* SunOS release 4.x clients.

Using Host Manager to Add SunOS Release 4.x Diskless/Dataless Clients

Host Manager makes it relatively simple to install clients as long as you have the information you need before you start. Table 10-1 lists this information.

Table 10-1 Information Needed to Add Network Clients

Client Type	The client's configuration type (standalone, diskless, or dataless)
Host Name	The client's host name
IP Address	The client's IP address
Ethernet Address	The client's Ethernet address
Timezone Region	The major geographic time zone region
Timezone	The local time zone
File Server	The client's file server
Media Server	A system from which to access software remotely
OS Release	The client's architecture and operating system version
Root/Swap Paths	Path names for the client's root file system and swap area
Swap Size	The swap size (in megabytes) for a diskless client
Terminal Type	<i>For SunOS 4.1.x systems only.</i> Provides a field to specify the terminal type for a diskless client

If you are setting up a Solaris 2.4 server for SunOS release 4.x clients, run the programs in the SUNWhinst package before you add clients with Host Manager.

When you display Host Manager from the Administration Tool, choose a file server and select the proper SunOS release 4.x version from the OS Release pull-down menu in the Add Host pop-up window.

Setting Up and Using Printers

11 

This chapter describes how to set up and administer printers after you install Solaris 2.4 software. This chapter also describes the changes to printer commands that have taken place between the SunOS release 4.x and the Solaris 2.4 release.

<i>Summary of Printing Differences</i>	<i>page 121</i>
<i>The Administration Tool's Printer Manager</i>	<i>page 122</i>
<i>Setting Up a Printer</i>	<i>page 124</i>
<i>Print Commands and the Compatibility Package</i>	<i>page 124</i>
<i>Using Printer Commands</i>	<i>page 125</i>

User Accounts, Printers, and Mail Administration and *Peripherals Administration* describe the SunOS release 5.4 setup and installation procedures and user commands for these devices.

Summary of Printing Differences

The SunOS release 5.4 LP print service replaces the SunOS release 4.x printing facilities, which were provided by the `lpd` daemon and `lpr`, `lpg`, `lprm`, and `lpc` commands. The Printer Manager in the Administration Tool 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

5.4 printers. For detailed documentation of the Printer Manager and the command-line interface to the LP service see *User Accounts, Printers, and Mail Administration*.

The services provided by the `/etc/printcap` file in SunOS release 4.x are handled in the Solaris 2.4 operating environment by the terminfo database and by the files in the `/etc/lp` directory.

The Administration Tool's Printer Manager

The Administration Tool's Printer Manager simplifies printer setup for local printers (or print servers and remote printing clients) by allowing you to avoid SunOS release 5.4 LP print service commands.

Because the Printer Manager is a distributed application, you can use it from one system to add, modify, or delete printers on other systems as long as they also have the Printer Manager installed. You cannot set up a SunOS release 4.x system or a non-Solaris system as a print server or print client using Printer Manager, but you can set up a Solaris 2.4 system, which has the Printer Manager, as a print client of a SunOS release 4.x print server.

The Printer Manager is a distributed Administration Tool application. Its graphical user interface guides you in setting up print servers and print clients as depicted in Figure 11-1.

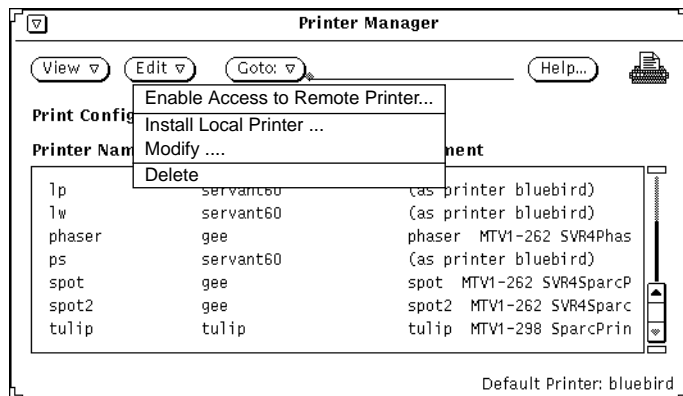


Figure 11-1 Printer Manager Functions

The Printer Manager has two main functions:

- Adds access to a remote printer
- Adds local printer

You can use the Printer Manager with the following configurations:

- SunOS release 5.x¹ print clients with a SunOS release 5.x¹ print server
- SunOS release 5.x¹ and SunOS release 4.x print clients with a SunOS release 5.x¹ print server
- SunOS release 5.x¹ and SunOS release 4.x print clients with a SunOS release 4.x print server

Note – If you want to administer a printer using the Administration Tool, ensure that you have an entry in `/etc/group` and that you are in group 14. You can edit the file `/etc/group` to add the entry if you have permissions.

Table 11-1 and Table 11-2 list the information you need to have ready for setting up local and remote printers.

Table 11-1 Information Needed to Set up a Local Printer

Printer Name	A unique name for the printer (maximum of 14 characters)
Comment	A description of the printer
Printer Port	The port the printer is connected to
Printer Type	A designation for the type of printer that is used to identify the <code>terminfo</code> database entry containing the control sequences that initialize the printer
File Contents	The particular format of the file being printed without being filtered (as long as the correct printer software is available)
Fault Notification	The method of notifying the superuser in case of a printer error. Choices are: Write to superuser (in the console window), Mail to superuser, or None.
System Default	The default printer, the printer your print requests will be sent to if you do not specify a printer
User Access List	The print clients that can print to this printer. By default, all print clients have access to this printer

1. This is true for all SunOS 5.x systems except SunOS 5.0.

Table 11-2 Information Needed to Enable Access to a Remote Printer

Printer Name	The remote printer name
Print Server	The name of the system the remote printer is connected to
Comment	Description of the printer that helps you identify it
Print Server OS	Print server's operating system, which is either Solaris release 2.x (System V) or SunOS release 4.1.x (BSD)
Default System Printer	The printer your print requests will be sent to if you do not specify a printer

Setting Up a Printer

Set up a local printer using either the Administration Tool (if you can run OpenWindows) or the command-line interface. The Administration Tool is easier to use, but some people prefer to run the commands themselves. For detailed information about using the Administration Tool or the command-line interface to set up a local printer, See to *User Accounts, Printers, and Mail Administration*.

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 2.4 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 2.4 system, the commands create message files that are handled by the SunOS release 5.4 LP print service scheduler.

Solaris 2.4 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 11-3 shows the basic user print command equivalents.

Table 11-3 User Print Command Equivalents

SunOS Release 4.x	SunOS Release 5.4	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 5.4 Printer Administration Commands

This section describes differences between printer setup and administration on SunOS release 4.x and Solaris 2.4 systems. All the underlying system services described are available only in the Solaris 2.4 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 *User Accounts, Printers, and Mail Administration* for details.

Table 11-4 shows the command equivalents for setting up printing.

Table 11-4 Printer Administration, Setup, and File Equivalents

SunOS Release 4.x	SunOS Release 5.4	Function
lpc	lpadmin	Control line printer functions
/etc/printcap	terminfo database and /etc/lp/printers/ <i>printername</i> /*	File that defines printer functions
/var/spool	/var/spool/lp	Directory where printing system stores spool and lock files
Not available	lpmove	Move print queues between printers
lpc down	reject	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 2.4 operating environment, you must specify that you want the file printed by piping (|) the output to the `lp` command. Table 11-5 shows the SunOS release 5.4 `troff` commands.

Table 11-5 SunOS Release 5.4 `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 terminal and modem management through the Administration Tool's Serial Port Manager and it describes the Services Access Facility (SAF), a Solaris 2.4 tool for managing access to local and network system services, such as modems and terminals.

This chapter contains the following sections:

<i>Terminal and Modem Management</i>	<i>page 129</i>
<i>Service Access Facility (SAF)</i>	<i>page 130</i>
<i>SAF Functions and Related Programs</i>	<i>page 132</i>

User Accounts, Printers, and Mail Administration and *Peripherals Administration* describe the details of Solaris 2.4 setup and installation procedures for serial devices.

Terminal and Modem Management

The Administration Tool's Serial Port Manager 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. To learn more about using the Serial Port Manager, see *Peripherals Administration* and *Administration Application Reference Manual*.

The Serial Port Manager 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 2.4 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 *Peripherals Administration*.

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 133</i>
<i>Changes to NFS</i>	<i>page 134</i>
<i>PPP</i>	<i>page 134</i>
<i>UUCP</i>	<i>page 134</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 5.4 TCP/IP facilities, see *TCP/IP Network Administration Guide* for complete information.

Changes to NFS

The Solaris 2.4 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 2.4 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 Network Administration Guide* for detailed information.

UUCP

The Solaris 2.4 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 2.4 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 2.4 UUCP that were not part of the SunOS release 4.x implementation. Table 13-2 on page 137 describes the log files added to Solaris 2.4 UUCP.

Table 13-1 New SunOS Release 5.4 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.
/etc/uucp/Grades	Maps text grade names to system names.
/etc/uucp/Limits	Specifies the number of concurrent UUCP sessions that can occur. Replaces <i>Maxuuscheds</i> and <i>Maxuuxqts</i> files in previous versions.
/etc/uucp/Config	Contains information to override UUCP parameters that can be tuned. Currently, the only parameter of this type is <i>Protocol</i> , so system administrators normally will not have to modify this file.
uuglist	Prints the list of service grades available on the system to use with the <i>-g</i> option of <i>uucp(1C)</i> and <i>uux(1C)</i> .

Solaris 2.4 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 2.4 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 5.4 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 2.4 operating environment. Solaris 2.4 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 2.4 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 5.4 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 5.4 UUCP, see *TCP/IP Network Administration Guide* for complete information.

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 140</i>
<i>NIS+</i>	<i>page 140</i>
<i>DNS</i>	<i>page 140</i>
<i>DNS and NIS+ Comparison</i>	<i>page 141</i>
<i>NIS and NIS+ Comparison</i>	<i>page 142</i>
<i>Planning an NIS+ Upgrade</i>	<i>page 143</i>
<i>Installing NIS+ on a SunOS Release 4.x System</i>	<i>page 143</i>

For more information about planning an NIS+ upgrade and installing NIS+, see *NIS+ Transition Guide*, *Name Services Configuration Guide*, and *Name Services Administration Guide*.

Note – The system administration documentation set for the Solaris 2.4 operating environment emphasizes a system that is using NIS+.

Name Service Switch

The Solaris 2.4 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 2.4 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 UNIX-style access rights for objects and entries
API and human interface	Allows read-only access to name service	Allows read-write access to name service. Provides: - 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: - 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 5.4

The Solaris 2.4 operating environment contains support for most client-side NIS utilities and APIs. There is *no* server-side support for NIS in the Solaris 2.4 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 2.4 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 5.4 installed on all servers and clients
- SunOS release 5.4 installed on one server, but combined with some SunOS release 4.x servers
- SunOS release 5.4 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 2.4 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 2.4 software, you can still use NIS+ to manage a heterogeneous network of SunOS release 4.x and Solaris 2.4 systems.

A `tar` archive file, `NISPLUS.TAR`, is included in the root (`/`) directory of the Solaris 2.4 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 5.4 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. NIS+ is also better integrated with other Solaris 2.x tools such as the Administration Tool.

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 2.4 operating environment. These changes affect all developers to varying degrees. The operating system kernel and its interfaces have also changed significantly since the SunOS release 4.x. This part of the guide describes these differences, points out similarities between releases, provides information you need to port existing drivers, or to write new ones for the Solaris 2.4 operating environment, and explains the implications for your programming environment.

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

<i>Compilers</i>	<i>page 147</i>
<i>Linkers</i>	<i>page 154</i>
<i>Debuggers</i>	<i>page 162</i>

Compilers

The single most significant change for developers migrating from SunOS release 4.x to the Solaris 2.4 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 added to 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 2.4 executables. `lint` and the `lint` libraries are also unbundled.

Solaris 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 2.4 ANSI C compiler. It is provided for your information, and is not intended to describe the range of operations available in default Solaris 2.4 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 ANSI C Solaris 2.4 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)	ANSI C (Solaris 2.4)	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 5.4)
-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)	ANSI C (Solaris 2.4)	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)	ANSI C (Solaris 2.4)	Description
-KPIC	-PIC	-PIC	Yes	Positions 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)	ANSI C (Solaris 2.4)	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)	ANSI C (Solaris 2.4)	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)	ANSI C (Solaris 2.4)	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 2.4 `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 2.4 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 2.4 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 156.
-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 156.)
-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 (only effective for -l options it precedes)
-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 2.4 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 2.4 operating environment. The -G option is now required to build a shared library. In SunOS release 4.x, the linker could infer that a shared library was being built. Now the linker must be explicitly instructed.

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 2.4 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 2.4 commands that can be used to create similar executables.

```
sunos4.1% ld -Bstatic test.o -lx           # uses libx.a and creates a static executable
sunos5.1% ld -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
sunos5.1% ld -dy 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
sunos5.1% ld -dy 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 2.4 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 2.4 Linker Search Paths With `LD_LIBRARY_PATH=dirlist1`

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

Table 15-6 Solaris 2.4 Linker Search Paths With `LD_LIBRARY_PATH=dirlist1,dirlist2`

	Library Search Path
	LD_LIBRARY_PATH= <i>dirlist1</i> , <i>dirlist2</i>
Link Editor	<i>dirlist1</i> , -L, <i>dirlist2</i> , /usr/ccs/lib, /usr/lib
Runtime Linker	<i>dirlist1</i> , <i>dirlist2</i> , -R, LD_RUN_PATH, /usr/lib

Version Numbering

SunOS release 4.x supported both major and minor version numbers on shared libraries. The Solaris 2.4 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 2.4 shared libraries.

Table 15-7 Example Shared Libraries

SunOS release 4.x	Solaris 2.4
<code>libc.so.1.7</code>	<code>libc.so.1</code>
<code>libdl.so.1.0</code>	<code>libdl.so.1</code>

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 this release, even though major numbers are still supported, the link editor does not match any version numbers. Using the previous example, the build environment link editor now searches for `libdl.so`.

If you want to support versioning, use the `-h` option when building a shared library. The information provided with the `-h` option is stored in the executable and used at runtime. This allows you to have multiple versions of a library and guarantees that the executable will use the version specified with the `-h` option when run.

If you do not use the `-h` option, the library name stored in the executable will not contain a version number.

Symbolic links have been created for most libraries in this release. Build any new shared libraries with major numbers, then create a symbolic link.

Examples

A new utility, `dump(1)` (See “Backing Up and Restoring Files” on page 105), 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 2.4 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 debugging under the Solaris 2.4 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 one significant improvement over the SunOS release 4.x `trace(1)` command: the ability to follow forked 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.

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 165</i>
<i>ptrace() Request Values</i>	<i>page 168</i>
<i>Libraries</i>	<i>page 170</i>
<i>Using make</i>	<i>page 172</i>
<i>Using SCCS</i>	<i>page 173</i>
<i>Determining Application Compatibility</i>	<i>page 173</i>
<i>Packaging Applications</i>	<i>page 174</i>
<i>Toolkits</i>	<i>page 176</i>
<i>Finding SunOS Release 4.x Tools</i>	<i>page 176</i>

ioctl() Requests

All `ioctl`s related to `dkio`, `filio`, `mtio`, `sockio`, `streamio`, `termio`, and `termios` are supported in this release.

A few incompatibilities exist between the SunOS release 4.x `termios` structure and Solaris 2.4 `termios` structure. Solaris 2.4 `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 2.4 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	TIOCSBRK
TIOCCBRK	TIOCSETX	TIOCGETX
OTTYDISC	NETLDISC	NTTYDISC
TABLDISC	NTABLDISC	MOUSELDISC
KBLDISC		

Table 16-2 shows `ttycom` `ioctl` requests that are not in the Solaris 2.4 operating environment.

Table 16-2 Unsupported `ttycom` `ioctl` Requests

TIOCSCTTY	TIOCGETPGRP	TIOCSETPGRP
TIOCNOTTY	TIOCMSET	TIOCMGET
TIOCMBIS	TIOCMBIC	TIOCTCNTL
TIOCSIGNAL	TIOCSSOFTCAR	TIOCGSOFTCAR
TIOCISPACE	TIOCISIZE	TIOCCONS
TIOCPKT	TIOCUCNTL	TIOCSTI
TIOCOUTQ		

Table 16-3 shows the `ioctl`s supported in the Solaris 2.4 operating environment.

Table 16-3 `ioctl()` Support

<code>ioctl()</code>	Description
<code>DKIOCGPART</code>	These requests are replaced with <code>DKIOCGAPART</code> and <code>DKIOCSAPART</code> in Solaris 2.4 software.
<code>DKIOCGCONF</code>	This request is replaced with <code>DKIOCINFO</code> in Solaris 2.4 software, which includes the combined information of the SunOS release 4.x <code>DKIOCGCONF</code> and <code>DKIOCINFO</code> structures.
<code>DKIOCSCMD</code>	This request succeeds only for XD, XY, and IPI drives. This <code>ioctl</code> fails for SCSI devices. Use the <code>USCSI ioctl</code> for SCSI devices.
<code>DKIOCGLOG</code>	<code>EINVAL</code> is returned. <code>DKIOCWCHK</code> toggles the write check on the floppy device.
<code>filio</code>	The following <code>filio ioctl</code> requests are not supported in this release or SVR4: <code>FIOSETOWN</code> , <code>FIOGETOWN</code> , <code>FIOCLEX</code> , <code>FIONCLEX</code> . <code>filio ioctl</code> requests not defined in the ABI or SVID.
<code>mtio</code>	<code>mtio ioctl</code> requests are not supported in Solaris 2.4 software, the ABI, SVID, or SVR4.
<code>sockio</code>	The following <code>sockio ioctl</code> requests are implemented in SVR4 and Solaris 2.4 software: <code>SIOCSPGRP</code> , <code>SIOCGPGRP</code> , <code>SIOCCATMARK</code> . <code>sockio ioctl</code> requests are not defined in the ABI or SVID.

Table 16-3 ioctl() Support (Continued)

ioctl()	Description
streamio	All SunOS release 4.x streamio ioctl requests are implemented in Solaris 2.4 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 2.4 software. Additionally, in Solaris 2.4 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 2.4 software, the ABI, SVID, and SVR4. A few incompatibilities exist between the SunOS release 4.x termios structure and Solaris 2.4 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 2.4 software, the ABI, SVID, or SVR4.

ptrace() Request Values

The ptrace() routine in Solaris 2.4 software 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 2.4 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 2.4 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 2.4 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 2.4 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 2.4 `ptrace()` routine.

Table 16-4 `ptrace()` Requests not Supported by Solaris 2.4 Software

PTRACE_ATTACH	PTRACE_GETWINDOW
PTRACE_DETACH	PTRACE_SETWINDOW
PTRACE_GETREGS	PTRACE_22
PTRACE_SETREGS	PTRACE_23
PTRACE_GETFPREGS	PTRACE_26
PTRACE_SETFPREGS	PTRACE_27
PTRACE_READDATA	PTRACE_28
PTRACE_WRITEDATA	PTRACE_SYSCALL
PTRACE_READTEXT	PTRACE_DUMPCORE
PTRACE_WRITETEXT	PTRACE_SETWRBKPT
PTRACE_GETFPAREGS	PTRACE_SETACBKPT
PTRACE_SETFPAREGS	PTRACE_CLDR7

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.

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 2.4 software.

Resource Limits

The Solaris 2.4 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_RSS	Maximum size (in bytes) to which a process's resident size 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 2.4 libraries and their locations.

Table 16-6 Comparison of Library Locations

Library Name	SunOS release 4.x Directory	Solaris 2.4 Directory
<code>libbsdmalloc.a</code>	<code>/usr/lib</code>	Not available
<code>libc.a</code>	<code>/usr/lib, and /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, and /usr/5lib</code>	<code>/usr/ucblib, and /usr/ccs/lib</code>
<code>libcurses_p.a</code>	<code>/usr/5lib</code>	Not available
<code>libdbm.a</code>	<code>/usr/lib</code>	<code>/usr/ucblib</code>
<code>libdl.so.1.0</code>	<code>/usr/lib</code>	<code>/usr/lib</code>
<code>libg.a</code>	<code>/usr/lib</code>	Not available
<code>libkvm.a</code>	<code>/usr/lib</code>	<code>/usr/lib</code>
<code>libkvm.so.0.3</code>	<code>/usr/lib</code>	<code>/usr/lib</code>
<code>libl.a</code>	<code>/usr/lib</code>	<code>/usr/ccs/lib</code>
<code>libln.a</code>	<code>/usr/lib</code>	Not available
<code>liblwp.a</code>	<code>/usr/lib</code>	Not available
<code>libm.a</code>	<code>/usr/lib</code>	<code>/usr/ccs/lib, and /usr/ccs/libp</code>
<code>libmp.a</code>	<code>/usr/lib</code>	<code>/usr/lib</code>
<code>libnbio.a</code>	<code>/usr/lib</code>	Not available

Table 16-6 Comparison of Library Locations (Continued)

Library Name	SunOS release 4.x Directory	Solaris 2.4 Directory
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
libxgl.so.1.1	/usr/lib	Not available
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 2.4 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 5.4 Makefile:
install:
    install -u bin -g bin -m 444 target.c /usr/bin/target
```

Check the compatibility tables in Appendix A, “Commands Reference Table,” for information about individual interfaces.

Using SCCS

The Solaris 2.4 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 2.4 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 2.4 operating environment.

One difference between SunOS release 4.x and Solaris 2.4 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 2.4 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 *Solaris 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.

Packaging Applications

The Solaris 2.4 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 2.4 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 2.4 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 2.4 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>

Table 16-10 Bundled Programming Tools (Continued)

SunOS release 4.x Command	SunOS release 4.x Location	Solaris 2.4 Location
cpp	/usr/lib/cpp	/usr/ccs/lib
delta	/usr/sccs	/usr/ccs/bin
error	/usr/ucb	/usr/ccs/bin
get	/usr/sccs	/usr/ccs/bin
help	/usr/sccs	/usr/ccs/bin
ld	/usr/bin	/usr/ccs/bin
lex	/usr/bin	/usr/ccs/bin
lorder	/usr/bin	/usr/ccs/bin
m4	/usr/bin	/usr/ccs/bin
make	/usr/bin	/usr/ccs/bin
nm	/usr/bin	/usr/ccs/bin
prof	/usr/bin	/usr/ccs/bin
prs	/usr/sccs	/usr/ccs/bin
prt	/usr/sccs	/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

Table 16-10 Bundled Programming Tools (Continued)

SunOS release 4.x Command	SunOS release 4.x Location	Solaris 2.4 Location
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 2.4 commands and their descriptions.

Table 16-11 New Programming Tools

New Command	Description
dis	Object code disassembler for COFF
dump	Dumps selected parts of an object file
exstr	Extracts strings from source files
mcs	Manipulates the comment section of an object file
regcmp	Regular expression compiler
truss	Traces system calls and signals

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

Table 16-12 Unbundled Programming Tools

Unbundled Command	Description
<code>gprof</code>	Displays call-graph profile data
<code>indent</code>	Indents and formats C program source files
<code>inline</code>	In-line procedure call expander
<code>lint</code>	A C program verifier
<code>objdump</code>	Dumps selected parts of a COFF object file
<code>ranlib</code>	Converts archives to random libraries
<code>tcov</code>	Constructs test coverage analysis and statement-by-statement profile
<code>trace</code>	Traces system calls and signals

Networking and Internationalization

17 

This chapter discusses Solaris 2.4 networking features as they relate to the programming environment, and it discusses issues concerning the improved internationalization features.

<i>Networking</i>	<i>page 181</i>
<i>Internationalization</i>	<i>page 183</i>

Networking

The Solaris 2.4 operating environment includes the following networking features:

- Distributed file system (DFS), which centralizes the file system utilities
- Network information services plus (NIS+)
- Name service switch file

See *Name Services Administration Guide* for more information on using these services.

NIS, NIS+

The Solaris 2.4 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 5.4” on page 143 for the nature of NIS support available in the Solaris 2.4 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 2.4 operating environment. Now Ethernet drivers are real STREAMS drivers that can be opened and communicated with directly.

The Solaris 2.4 Ethernet drivers and other data link drivers support the connectionless Data Link Provider Interface (DLPI) Version 2 specification.

Sockets

Sockets are not defined as part of the ABI, although they are supported in the Solaris 2.4 operating environment. Unlike the SunOS release 4.x, sockets are no longer a kernel-level interface. They are now a library, *libsocket*, implemented on STREAMS.

Internationalization

Developers concerned with internationalization will find that most of the changes in the Solaris 2.4 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 `srftime(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 2.4 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 2.4 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 2.4 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 2.4 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 2.4 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 2.4 library).

Table 17-1 shows the new location of these interfaces.

Table 17-1 xpg2lib Library Routine Locations

Routine	SunOS 4.x Location	Solaris 2.4 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 2.4 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 2.4 operating environment that affect kernel and system developers.

This chapter contains the following sections:

<i>System Configuration</i>	<i>page 187</i>
<i>Device Configuration: The Reconfiguration Boot</i>	<i>page 191</i>
<i>Device Naming From a Developer's Perspective</i>	<i>page 191</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 `/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, `/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*.

The `/kernel/drv` directory is used for bootable devices and code that is specific to the system on which it runs. The `/usr/kernel/drv` directory is used for all other drivers.

`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 2.4 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 2.4 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 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
- `inetboot` - To boot from across the network
- `hsfsboot` - To boot from a CD-ROM

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/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 (`/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. In theory, the `/ufsboot` code would 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 *File System Administration* 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 2.4 operating environment.

Table 18-1 Summary of Boot Differences

SunOS release 4.x	Solaris 2.4	Description
boot block	bootblk	Loads <code>ufsboot</code> from disk
boot program	ufsboot	Loads <code>unix</code> from disk
vmunix	unix	Bootable kernel image
boot.sun4c.sunos.4.1.1	inetboot	Mounts and copies <code>unix</code> from network
rc.boot, rc.single	/etc/rcS	Mounts <code>/usr</code> 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
```

Or, on a Sun-4 system, using the following.

```
>b -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
# /etc/init.d/drvconfig
# /etc/init.d/devlinks
```

Device Naming From a Developer's Perspective

This section expands on the discussion in “Device Naming Conventions” on page 65, 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 191. `drvconfig` configures `/devices` with information about devices (with drivers) that are connected and turned on at startup 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 exact 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)`, `xy(7)`, and `xd(7)` 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 (partition "g"))
```

or an `sd` device, such as:

```
/dev/dsk/c0t3d0s0 (controller 0, target 3, lun 0, slice 0 (partition "a"))
```

The logical name conveys nothing definite 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 *Peripherals Administration*.

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 2.4 driver architecture.

<i>Device Drivers and STREAMS Device Drivers</i>	<i>page 195</i>
<i>Device Driver Commands</i>	<i>page 203</i>

See the following guides for more information on the topics discussed in this chapter:

- *Writing Device Drivers*
- *STREAMS Programmer's Guide*
- *System Services Guide*
- *Peripherals Administration*

Device Drivers and STREAMS Device Drivers

Some of the many changes to device drivers in the Solaris 2.4 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 2.4 SPARC DDI/DKI is to provide binary compatibility of device drivers across all supported platforms and for all future releases of the Solaris 2.4 operating environment on those platforms.

In the Solaris 2.4 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 2.4 operating environment are formalized and are referred to as the *Solaris 2.4 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 drivers for Solaris 2.4 environments can be treated like any other consumer software product.

In the Solaris 2.4 DDI/DKI the DDI-only interfaces are generic to all SPARC-based systems that support the Solaris 2.4 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 2.4 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 2.4 DDI/DKI compliant*. A Solaris 2.4 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

`devinfo` *Command*

The Solaris 2.4 `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 2.4 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 2.4 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.

Before you determine how you will support the driver in the Solaris 2.4 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

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.

`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,
 - `dev_t`
 - `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 `strioctl` structure and then use:

```
ioctl(fd, I_STR, &strioctl);
```

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 2.4 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 Programmer's 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 2.4 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 Programmer's 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, framebuffer, 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*. Currently, the Solaris 2.4 DDI/DKI supports only the writing of 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.

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 2.4 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 2.4 command is provided.
- If an interface is listed as “the same” (S), the Solaris 2.4 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 5.4 column.

Note – The directory structure in the SunOS release 5.4 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

≡ A

`/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 2.4 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 5.4	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 SunOS release 5.4 base release. This command is available if you install the SunOS/BSD Compatibility package on your SunOS release 5.4 system. The `init 0` command replaces `fasthalt` in SunOS release 5.4. 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/uch` 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 5.4	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 5.4. 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 148 for details.

Table A-3 Example 3

SunOS Release 4.x Command	SunOS Release 5.4	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 5.4. 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 5.4 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 5.4. If you are accustomed to using SunOS release 4.x /usr/bin/date command, you should look at the SunOS release 5.4 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 5.4	Alternative Available and Notes	BSD
rev(1)	N		N

The SunOS 4.2 rev command is not available in the SunOS release 5.4 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 2.4 environment and in the SunOS/BSD Source Compatibility Package.

≡ A

Table A-5 Commands Reference Table

SunOS Release 4.x Command	SunOS Release 5.4	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)	mailx	The mailx(1) command provides similar capabilities.	N
ac(8)	sar	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 5.4	Alternative Available and Notes	BSD
admin(1)	C	The following SunOS release 4.x options are not available in the SunOS release 5.4 system software: -b Force encoding of binary data. -l <i>release</i> [, <i>release</i> . . .] Lock indicated release against deltas.	N
adv(8)	N	RFS does not exist.	N
aedplot(1G)	N		S
align_equals(1)	OpenWindows	The OpenWindows text menu indent command provides similar capabilities.	N
analyze(8)	adb	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 5.4, this database is called <code>windex</code> , and the format is slightly different.	N
ar(1V)	C	The <code>a</code> , <code>b</code> , and <code>i</code> options have similar capabilities in the SunOS release 5.4 system software. The following SunOS release 4.x modifiers are not supported in SunOS release 5.4: -a <i>position-name</i> Place new files after <i>position-name</i> . -b <i>position-name</i> Place new files before <i>position-name</i> . -i <i>position-name</i> Place new files before <i>position-name</i> . -u Used with the <code>r</code> option to replace only those files which have changed since they were archived. -v Verbose. See the <code>ar(1V)</code> manual page for more information.	N
ar(1V) -- SysV	C		N
arch(1)	uname	Similar capabilities are available with <code>uname -p</code> . SunOS release 4.x shell scripts used the <code>arch</code> command to determine system architecture. Use <code>uname -m</code> (kernel architecture) or <code>uname -p</code> (processor) as a replacement in SunOS release 5.4 scripts.	S
arp(8C)	S		N
as(1)	C	The following SunOS release 4.x options are not available in the SunOS release 5.4 command: -d2, -h, -j, -J, -k, -L, -O[<i>n</i>] -P.	N

≡ A

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
at(1)	S	The at, atq, and atrm commands in SunOS release 5.4 systems behave slightly differently than they do in SunOS release 4.x systems. Security for non-privileged users is more restricted on SunOS release 5.4 systems. Non-privileged users cannot display the jobs of any other user.	N
atoplot(1G)	N		S
atq(1)	C	The at, atq, and atrm commands in SunOS release 5.4 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 5.4 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 5.4 systems.	N
atrm(1)	C	The at, atq, and atrm commands in SunOS release 5.4 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 5.4 command. Security for non-privileged users is more restricted on SunOS release 5.4 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 5.4 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 5.4. The default home directory path is /export/home/ <i>username</i> .	N
awk(1)	S		N
banner(1V) -- SysV	S		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
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
basename(1)	S	The SunOS release 5.4 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 5.4 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 5.4 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	When users log on, startup shell scripts often use the biff command to set default file protection for the user. Replace those commands to make SunOS release 5.4 scripts work correctly. The SunOS release 5.4 chmod o+x /dev/tty command is equivalent to biff -y, while chmod o-x /dev/tty is equivalent to biff -n.	S
bin-mail(1)	S	Same as the SunOS release 5.4 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

≡ A

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
cat(1V)	S	The SunOS release 5.4 <code>cat</code> command requires the <code>-v</code> option with the <code>-t</code> and <code>-e</code> options. The SunOS release 5.4 command displays FORMFEED characters with the <code>-t</code> option, instead of the <code>-v</code> option as with the SunOS release 4.x command.	N
catman(8)	S		N
cb(1)	S		N
cc(1V) -- SysV	N		N
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 5.4 command silently ignores the error.	N
cflow(1V) -- SysV	N	The <code>cflow</code> command is now available as an unbundled product.	N
cflow(1V)	N	The <code>cflow</code> command is now available as an unbundled product.	N
change_login(8)	N		N
chargefee(8)	S		N
check4(8)	N		N
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 5.4 system software. In SunOS release 4.x system software, <code>chgrp</code> changed ownership of the symbolic itself; in SunOS release 5.4 system software, <code>chgrp</code> follows the link. To change ownership of the symbolic link in SunOS release 5.4 system software, use the <code>-h</code> option.	N
chkey(1)	S		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
chmod(1V) -- SysV	C	The SunOS release 5.4 -R option changes the mode of the target when symbolic links are encountered.	N
chmod(1V)	S	The SunOS release 5.4 -R option changes the mode of the target when symbolic links are encountered. The SunOS release 5.4 command supports two additional permissions: 'l' and 't'.	N
chown(8)	C	The default behavior of symbolic links has changed. SunOS release 4.x chown changed ownership of the symbolic link. SunOS release 5.4 chown follows the link. To change the ownership of the link, use chown -h. The SunOS release 5.4 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
client(8)	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
cluster(1)	N		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

≡ A

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
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
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 5.4 command silently ignores the error.	N
comm(1)	S		N
compress(1)	S		N
config(8)	N		N
copy_home(8)	N		N
cp(1)	C	The -R option is replaced by the -r option in the SunOS release 5.4 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 5.4	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

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
<code>ctrace(1V) -- SysV</code>	N	The following SunOS release 4.x option is not available in the SunOS release 5.4 command: <code>-b</code> Use only basic functions to trace code. This option is needed for running under an operating system that does not have the <code>signal()</code> , <code>fflush()</code> , <code>longjmp()</code> or <code>setjmp()</code> functions available. The syntax of the <code>-r</code> option differs between SunOS release 4.x and SunOS release 5.4 system software. The 4.1 format is <code>-rf</code> ; in SunOS release 5.4, it is <code>-r f</code> . <code>ctrace</code> is available as an unbundled product.	N
<code>cu(1C)</code>	S		N
<code>cut(1V) -- SysV</code>	S		N
<code>cxref(1V) -- SysV</code>	S		N
<code>cxref(1V)</code>	N	<code>cxref</code> is available as an unbundled product.	N
<code>date(1V) -- SysV</code>	S		N
<code>date(1V)</code>	C	The format used when setting the date is slightly different in SunOS release 5.4. See the <code>date(1)</code> man page for more information.	N
<code>dbconfig(8)</code>	S		N
<code>dbx(1)</code>	N	Available with the unbundled SPARCworks product.	N
<code>dbxtool(1)</code>	N	Available with the unbundled SPARCworks product as the command debugger.	N
<code>dc(1)</code>	S		N
<code>-dcheck(8)</code>	<code>fsck</code> and <code>ncheck</code>	Use the <code>fsck(1M)</code> command for normal consistency checking. The <code>ncheck(1M)</code> command replaces the function of <code>dcheck -i</code> numbers.	N
<code>dd(1)</code>	C	In the SunOS release 4.x command, the size used for the size suffix <code>w</code> (words) is in units of 4 bytes, while in SunOS release 5.4 system software, <code>w</code> is in units of 2 bytes. <code>k</code> , <code>b</code> , or <code>w</code> may be used as a suffix to specify multiplication by 1024, 512, or 2, respectively. The <code>unblock</code> and <code>block</code> conversion options are new with SunOS release 5.4.	N

≡ A

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
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
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 5.4 command.	N
deroff(1)	S		N
des(1)	S		N
devinfo(8S)	C	The <code>sysdef</code> command provides similar capabilities.	N
devnm(8)	C	The output format between SunOS release 4.x and SunOS release 5.4 system software is quite different. In SunOS release 4.x system software, the name argument is optional. In the SunOS release 5.4 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 5.4 <code>df</code> command. The SunOS release 5.4 <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 5.4 <code>df -t</code> command prints full listings with totals. You can use <code>df -l</code> to see local filesystems.	S

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
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 5.4 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
dis(1)	C	The following SunOS release 4.x option is not available in the SunOS release 5.4 command: <code>-da sec</code> Disassemble <code>sec</code> as data, printing the actual address of the data. Use the SunOS release 5.4 <code>-D sec</code> option to do the same thing.	N
disablenumlock(1)	N		N
diskusg(8)	A	The <code>acctdusg (1M)</code> command provides similar capabilities.	N
dkctl(8)	N		N
dkinfo(8)	prvtoc	The <code>prvtoc(1M)</code> 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
dos(1)	N		N
dos2unix(1)	S		N
du(1V) -- SysV	S		N

≡ A

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
du(1V)	C	The SunOS release 4.x command reports the disk usage in kilobytes while the SunOS release 5.4 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)	ufsdump	The ufsdump command provides similar capabilities. The following SunOS release 4.x options are not in the SunOS release 5.4 command: -a <i>archive-file</i> The SunOS release 5.4 -a option dumps the archive header of each member of an archive. -D Specify diskette as the dump media. The SunOS release 5.4 -D option dumps debugging information. -v Verify against the file system being dumped. The SunOS release 5.4 -v option dumps information in symbolic, rather than numeric, representation.	N
dumpfs(8)	fstyp	The fstyp -F ufs -v command provides similar capabilities.	N
dumpkeys(1)	S		N
e(1)	ex	The ex(1) command provides similar capabilities.	N
echo(1V) -- SysV	S		N
echo(1V)	C	The -n option suppressed new-line printing under SunOS release 4.x. Use a \c for SunOS release 5.4.	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
enblenumlock(1)	N		N
enroll(1)	N		N
env(1)	S		N
eqn(1)	S		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
error(1)	S		N
etherd(8C)	snoop	The snoop(1M) command provides similar capabilities.	N
etherfind(8C)	snoop	The snoop(1M) command provides similar capabilities.	N
ex(1)	S		N
expand(1)	S		N
exportfs(8)	A	The share(1M) command provides similar capabilities.	N
expr(1V) -- SysV	S		N
expr(1V)	C		S
extract_files(8)	A	The pkgadd(1M) command provides similar capabilities.	N
extract_patch(8)	A	The pkgadd(1M) command provides similar capabilities.	N
extract_unbundled(8)	A	The swmtool(1M) command provides similar capabilities.	N
false(1)	S		N
fastboot(8)	A	The init 6 command provides similar capabilities.	S
fasthalt(8)	A	The init 0 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 5.4 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 5.4 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

≡ A

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
fontedit(1)	N		N
fontflip(1)	S		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)	prtconf		N
from(1)	N		S
fsck(8)	C	The SunOS release 4.x <code>fsck</code> command differs significantly from the SunOS release 5.4 command. With the SunOS release 5.4 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		N
fusage(8)	S		N
fuser(8)	S		N
fwtmp(8)	S		N
gcore(1)	S		N
generic_args(1)	N		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
get(1)	C	The SunOS release 5.4 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 5.4 command ignores them silently.	N
get_alarm(1)	N		N
get_selection(1)	xv_get_sel	The xv_get_sel(1) 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
gid_allocd(8C)	N		N
gigiplot(1G)	N		S
glob(1)	S		N
goto(1)	S		N
gpconfig(8)	N		N
gprof(1)	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

≡ A

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
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
help_viewer(1)	N		N
hostid(1)	sysdef	The <code>sysdef -h</code> command provides similar capabilities.	S
hostname(1)	uname	The <code>uname -n</code> command provides similar capabilities.	S
hostrfs(8)	N	RFS is not available.	N
hp7221plot(1G)	N		S
hpplot(1G)	N		S
htable(8)	S		N
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
imemtest(8C)	N		N
implot(1G)	N		
in.comsat(8C)	S		N
in.fingerd(8C)	S		N
in.ftpd(8C)	S		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
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 5.4 commands. In SunOS release 4.x system software, the range is 512-1023; in SunOS release 5.4 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
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 5.4 system software. In the SunOS release 5.4 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 5.4 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		N
input_from_defaults(1)	N		N

≡ A

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
insert_brackets(1)	OpenWindows	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 5.4 commands.	S
installboot(8S)	S		N
installtxt(8)	msgfmt	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
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 5.4 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 5.4 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

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
lastlogin(8)	S		N
ld(1)	C	There are many differences between the SunOS release 4.x ld command and the SunOS release 5.4 command. The following SunOS release 4.x options are not available with SunOS release 5.4: <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 5.4 by the <code>-z</code> option. The <code>-d</code> , <code>-dc</code> , <code>-dp</code> options are the default in SunOS release 5.4 system software. To turn off these options use <code>-b</code> .	S
ldconfig(8)	N		N
ldd(1)	S		N
leave(1)	cron	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 5.4 command: <code>-f</code> Compile faster by not packing resulting tables. This option is limited to small programs.	N
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 ln command never removes the target if it already exists. The SunOS release 5.4 ln 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 5.4 system software.	N
load(1)	A	The <code>swmtool(1M)</code> command provides some of the capabilities of the <code>load</code> command.	N

≡ A

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
load_package(8)	A	The <code>swmtool(1M)</code> command provides some of the capabilities of the <code>load_package</code> command.	N
loadc(1)	A	The <code>swmtool(1M)</code> command provides part of the capabilities of the SunOS release 4.x <code>loadc</code> command.	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
logintool(8)	N		N
logname(1)	S		N
look(1)	S		N
lookbib(1)	S		N
lorder(1)	S		N
lp(1)	S		N
lpc(8)	lpadmin	The <code>lpadmin(1M)</code> command provides similar capabilities.	S
lpd(8)	lpadmin	The <code>lpadmin(1M)</code> command provides similar capabilities.	S
lpq(1)	lpstat	The <code>lpstat(1)</code> command provides similar capabilities.	S
lpr(1)	lp	The <code>lp(1)</code> command provides similar capabilities.	S
lprm(1)	cancel	The <code>cancel(1)</code> 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

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
m4(1V)	S		N
m4(1V) -- SysV	S		N
m68k(1)	S		N
mach(1)	uname	The <code>-p</code> option of the <code>uname(1)</code> command provides similar capabilities.	S
machid(1)	N		S
mail(1) -- UCB	mailx		S
mail(1)	C	Now in <code>/usr/bin/mail</code> , was in <code>/usr/ucb/mail</code> in SunOS release 4.x. This entry refers to the <code>mail</code> command installed under <code>/usr/bin/mail</code> . The SunOS release 4.x <code>mail</code> is compatible with the SunOS release 5.4 command except for the following: <code>-i</code> The <code>-i</code> (ignore interrupts) option is not available with SunOS release 5.4. <code>postmark</code> line In the SunOS release 4.x command, the <code>postmark</code> line is preceded by a <code>></code> ; this is not required by the SunOS release 5.4 command. <code>mail forwarding</code>	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
make(1)	S	SVR4 & SVID <code>make</code> is available in <code>/usr/ccs/lib/svr4.make</code>	N
makedbm(8)	C	The SunOS release 5.4 interface for this command is compatible with the SunOS release 4.x interface. The SunOS release 5.4 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

≡ A

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
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 -s 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 5.4 command provides for different file system types.	N
mknod(8)	C	In the SunOS release 4.x command, only the superuser can create character and block special files. In the SunOS release 5.4 command, other users with appropriate privileges can too.	N
mkproto(8)	C		N
mkstr(1)	N		S
modload(8)	S		N
modstat(8)	modinfo	The <code>modinfo(1M)</code> command provides similar capabilities.	N
modunload(8)	S		N
monacct(8)	S		N
more(1)	S		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
mount(8)	C	The interface differs significantly between the two versions. In the SunOS release 5.4 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)	C	The SunOS release 4.x '-' option is not available in SunOS release 5.4 system software: this option allows a user to explicitly mark the end of any command line options. It allows mv to recognize file name arguments that begin with a'-'.	N
named(8C)	in.named		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
netconfig(8C)	N		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

≡ A

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	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 5.4 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 5.4, it happens immediately. In SunOS release 4.x, <code>addr</code> can be specified in hexadecimal notation while in SunOS release 5.4 it cannot. The SunOS release 4.x <code>-m</code> option is not available in the SunOS release 5.4 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 5.4 version: <code>-g</code> , <code>-p</code> , <code>-s</code> , and <code>-a</code> . The SunOS release 4.x and SunOS release 5.4 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
objdump(1)	C	The <code>dump(1)</code> command provides similar capabilities.	N
od(1V)	S		N
od(1V) -- SysV	S		N
old-analyze(8)	N		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	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
organizer(1)	N		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 5.4 system software, they are restricted to {NAME_MAX} - 2. The SunOS release 5.4 pack and unpack commands are compatible with the SunOS release 4.x commands.	N
page(1)	S		N
pagesize(1)	N		S

≡ A

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	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)	cpio	The <code>cpio(1)</code> command provides similar capabilities.	N
paxcpio(1V)	cpio	The <code>cpio(1)</code> command provides similar capabilities.	N
pcat(1V) -- SysV	S		N
pdp11(1)	S		N
perfmeter(1)	OpenWindows	This command is available in SunOS release 5.4 as the OpenWindows Performance Meter tool.	N
pg(1V) -- SysV	S		N
ping(8C)	S		N
plot(1G)	N		S
plottoa(1G)	N		S
pnps386(8C)	N		N
pnpsboot(8C)	N		N
pnpsd(8C)	N		N
portmap(8C)	rpcbind	The <code>rpcbind(1M)</code> command 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)	env	The <code>env(1)</code> command provides similar capabilities.	S

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
prof(1)	C	The SunOS release 4.x <code>-v</code> option is not available with SunOS release 5.4 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 5.4 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 5.4 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 5.4 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 5.4 command, it prints information in a format that reflects the new process scheduler design. <code>-e</code> In the SunOS release 4.x command, this option displays the environment as well as the arguments to the command.	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

≡ A

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
quotaoff(8)	S		N
quotaon(8)	S		N
ranlib(1)	ar	The ar(1) command provides similar capabilities.	N
rarpd(8C)	S		N
rasfilter8tol(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 5.4 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)	priocntl	The priocntl(1) command provides similar capabilities.	S
repquota(8)	S		N
reset(1)	A	stty provides similar capabilities.	S
restore(8)	A	The SunOS release 5.4 command, ufsrestore, has been enhanced to take advantage of the end-of-media detection done by ufsdump.	N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
rev(1)	N		N
rexd(8C)	A	in.rexd provides similar capabilities.	N
rexcd(8C)	A	in.rexcd 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 5.4 command. Also, the syntax for the -e option differs between the SunOS release 4.x and SunOS release 5.4 commands. In SunOS release 4.x system software, the syntax is -ec; in SunOS release 5.4 system software, it is -e c.	N
rlogind(8C)	in.rlogind		N
rm(1)	C	The SunOS release 4.x '-' option is not available in the SunOS release 5.4 command. This option allows a user to explicitly mark the end of any command line options. This option allows mv to recognize file name arguments that begin with a '-'. The SunOS release 5.4 mv command provides this capability using the '--' option (a double dash).	N
rm_client(8)	A	The admintool(1M) utility replaces this command on SunOS release 5.4 systems.	N
rm_services(8)	A	The swmttool(1M) command provides similar capabilities	N
rmail(8C)	C	The SunOS release 4.x version handles remote mail received using uucp(1C). It is explicitly designed for use with uucp(1C) and sendmail(8). The SunOS release 5.4 rmail is a link to mail(1) and is a command used for reading mail.	N

≡ A

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
rm (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 5.4 command silently ignores the error.	N
rmdir(1)	S		N
rmntstat(8)	N	RFS is not available.	N
rmt(8C)	S		N
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 5.4 command uses <code>gethostbyname(3)</code> .	N
routed(8)	in.routed		N
rpc.bootparamd(8)	S		N
rpc.etherd(8C)	N		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.ypupdated(8C)	N		N
rpcgen(1)	S		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
rpcinfo(8)	S		N
rrestore(8)	A	The ufsrestore(1M) command provides similar capabilities.	N
rsh(1C)	S		N
runacct(8)	S		N
rup(1C)	S		N
ruptime(1C)	S		N
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 5.4 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

≡ A

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
sccs-unget(1)	S		N
sccs-val(1)	S		N
sccsdiff(1)	C		N
screenblank(1)	OpenWindows	The OpenWindows <code>xset -s 600</code> command provides similar capabilities.	N
screendump(1)	N		N
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 <code>/usr/5bin/sed</code> and the SunOS release 5.4 commands do not strip initial SPACE and TAB characters from text lines.	S
selection_svc(1)	sv_xv_sel_svc	The <code>sv_xv_sel_svc(1)</code> command provides similar capabilities.	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 <code>echo</code> and <code>test</code> depend on the relative positions of <code>/usr/bin</code> and <code>/usr/5bin</code> in the environment variable <code>PATH</code> . No such dependency exists under SunOS release 5.4 system software.	N
shelltool(1)	OpenWindows	This command is available as an OpenWindows Shell Tool.	N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
shift_lines(1)	OpenWindows	An OpenWindows command is available with the OpenWindows Text Editor.	N
showfh(8C)	S		N
showmount(8)	S		N
shutacct(8)	S		N
shutdown(8)	C	The SunOS release 4.x command is very different from the SunOS release 5.4 <code>shutdown(1M)</code> command. By default, the SunOS release 5.4 <code>shutdown(1M)</code> asks for confirmation before starting shutdown activities, while the SunOS release 4.x <code>shutdown(8)</code> does not ask for confirmation. In addition, the following SunOS release 4.x options are not present in the SunOS release 5.4 command: <code>-f</code> , <code>-h</code> , <code>-k</code> , <code>-n</code> , <code>-r</code> , <i>time</i> , <i>warning-message</i> .	S
size(1)	C	The SunOS release 4.x command prints sizes in hexadecimal and decimal, and the file name is optional (with <code>a.out</code> as the default). The SunOS release 5.4 command prints them only in decimal, unless the <code>-o</code> or <code>-x</code> option is specified, and the file name is required.	N
skyversion(8)	N		N
sleep(1)	S		N
snap(1)	A	<code>admintool (1M)</code> provides similar capabilities.	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 <code>-h spellhist</code> option is not available with the SunOS release 5.4 command. This option places misspelled words with a user/date stamp in <i>spellhist</i> .	N
spellin(1)	S		N
spline(1G)	S		N

≡ A

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
split(1)	S		N
spray(8C)	C	The SunOS release 4.x- <i>i delay</i> option is not available with the SunOS release 5.4 command. This option specifies that ICMP echo packets should be used rather than RPC.	N
startup(8)	S		N
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 5.4 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 - <i>f</i> option is not supported by the SunOS release 4.x /usr/5bin/su or SunOS release 5.4 su command. This option is used for a fast su.	N
sum(1V) -- SysV	S		N
sum(1V)	C		S
sun(1)	S		N
sundiag(8)	S		N
suninstall(8)	C	The command to install SunOS release 5.4 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 5.4 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 5.4 swap command replace capabilities of individual swap-related commands, such as swapon, in SunOS release 4.x systems.	N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
swin(1)	N		N
switcher(1)	N		N
symorder(1)	S		N
sync(1)	S		N
sys-config(8)	admintool	The admintool(1M) utility replaces this command on SunOS release 5.4 systems.	N
sys-unconfig(8)	S		N
sysex(1)	N		N
syslogd(8)	S		N
syswait(1)	N		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)	C	The SunOS release 4.x talk command can be used to communicate with a user on another host. The SunOS release 5.4 command does not support remote communication with other hosts.	N
tar(1)	S		N
tbl(1)	S		N
tcopy(1)	S		N
tcov(1)	N	The lprof(1) command is available as an unbundled product and provides similar capabilities.	N
tee(1)	S		N
tek(1G)	N		S
tektool(1)	N		N

≡ A

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
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
tfsd(8)	N		N
tftp(1C)	S		N
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 5.4 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

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
true(1)	S		N
tset(1)	N	The SunOS release 4.x-S option is not available with the SunOS release 5.4 command. This option outputs strings containing suitable values for the TERM and TERMCAP environment variables.	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
tzsetup(8)	N		N
u370(1)	S		N
u3b(1)	S		N
u3b15(1)	S		N
u3b2(1)	S		N
u3b5(1)	S		N
uid_allocd(8C)	N		N
ul(1)	S		N
umask(1)	S		N
umount(8)	C	The interface differs significantly between the two versions. In the SunOS release 5.4 command, most options are changed and must be supplied as file system-specific options.	N
umount_tfs(8)	N		N
unadv(8)	N		N
uname(1)	S		N
uncompress(1)	S		N
unconfigure(8)	N		N

≡ A

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
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 5.4 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	Any user can unlink a directory.	N
unload(1)	A	The SunOS release 5.4 <code>swmtool (1M)</code> command provides some of these capabilities.	N
unloadc(1)	A	The SunOS release 5.4 <code>swmtool (1M)</code> command provides some of these capabilities.	N
unpack(1V) -- SysV	S		N
unpack(1V) -- SysV	C	With the SunOS release 4.x <code>/usr/5bin/pack</code> command, file names are restricted to 12 characters. In SunOS release 5.4 system software, they are restricted to <code>{NAME_MAX} - 2</code> . The SunOS release 5.4 <code>pack</code> and <code>unpack</code> commands are compatible with the SunOS release 4.x commands.	
unset4(8)	N		N
unwhiteout(1)	N		N
update(8)	<code>fsflush</code>	The <code>fsflush</code> command provides similar capabilities.	N
uptime(1)	<code>who</code>	The <code>who -b</code> command provides similar capabilities.	N
users(1)	<code>who</code>	The <code>who -q</code> provides similar capabilities.	S
ustar(1V)	<code>tar</code>	The <code>tar(1)</code> command provides similar capabilities.	N
uucheck(8C)	S		N
uucico(8C)	S		N
uucleanup(8C)	S		N
uucp(1C)	S		N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
uudecode(1C)	S		N
uuencode(1C)	S		N
uulog(1C)	C	The <code>-u</code> 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
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 <code>-f</code> option is no longer available.	N
vplot(1)	N		N
vswap(1)	N		N
vtroff(1)	N		N
vwidth(1)	N		N
w(1)	S		N

≡ A

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
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 <code>id(1)</code> command provides similar capabilities. The <code>id</code> command prints the user name and user and group IDs, instead of just the user name.	S
whois(1)	S		N
write(1)	S		N
xargs(1V) -- SysV	S		N
xget(1)	N		N
xsend(1)	N		N
xstr(1)	S		N
yacc(1)	S		N
yes(1)	N		N
ypbatchupd(8C)	N		N
ypbind(8)	S	Now in <code>/usr/lib/netsvc/yp</code> .	N
ypcat(1)	S		N
ypinit(8)	S		N
ypmatch(1)	S		N
yppasswd(1)	S	The <code>yppasswd</code> command is still available on SunOS release 5.4 systems to access the password information on NIS servers. The equivalent command for NIS+ databases is <code>nispasswd(1)</code> and the equivalent command for systems with no name service is <code>passwd(1)</code> .	N

Table A-5 Commands Reference Table (Continued)

SunOS Release 4.x Command	SunOS Release 5.4	Alternative Available and Notes	BSD
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
zcat(1)	S		N
zdump(8)	S		N
zic(8)	S		N

≡ A

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 release 2.4, 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 release 2.4 system call is provided.
- If an interface is listed as “the same” (S), the Solaris release 2.4 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.

≡ B

- 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 release 2.4 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 5.4	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 `memctl()` system call is not available in the ABI, SVID, SVR4, or the SunOS release 5.4. 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 5.4	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 5.4 version.	N	N	S#	N

The `getsockname()` system call is not defined in the ABI, or SVID. The `getsockname()` call in SunOS release 5.4 and SVR4 releases is the same as the one in SunOS release 4.x, except SunOS release 5.4 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 Release 5.4	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 5.4 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 5.4 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 5.4 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 5.4 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 5.4 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 5.4 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 5.4, 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 5.4, 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 Release 5.4	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 5.4, ABI, SVID, or SVR4 versions: S_IREAD (00400), S_IWRITE (00200), S_IEXEC (00100). However, the equivalent SunOS release 5.4, 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 5.4, 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 5.4 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 5.4 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 Release 5.4	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 5.4, ABI, SVID, and SVR4, the <i>mode</i> argument is of type <code>mode_t</code> . Also, SunOS release 5.4, 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 5.4, 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 5.4, 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 5.4, 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 5.4, 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 5.4, 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 5.4, 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 Release 5.4	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 5.4, 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 5.4, 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>FNDELAY</code> flags defined in < <code>sys/file.h</code> >. SunOS release 5.4, 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>FNDELAY</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_GETTOWN</code> and <code>F_SETTOWN</code> commands are not supported in SunOS release 5.4, 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 5.4, ABI, SVID, and SVR4, <code>fork()</code> returns a value of type <code>pid_t</code> . Also, SunOS release 5.4, 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 Release 5.4	Notes	ABI	SVID	SVR4	BSD
fstatfs(2)	A	The <code>fstatvfs(2)</code> system call provides equivalent functionality.	A	A	A	S
fsync(2)	S		S	S	S	N
ftruncate(2)	S	Now <code>ftruncate(3C)</code> .	N	N	S	N
getauid(2)	N		N	N	N	N
getdents(2)	S		N	N	S	N
getdirenties(2)	A	The <code>getdents(2)</code> system call provides equivalent functionality.	N	N	N	N
getdomainname(2)	A	The <code>sysinfo(2)</code> system call provides equivalent functionality.	N	N	N	N
getdtablesize(2)	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
getegid(2V)	C	In SunOS release 4.x, <code>getegid()</code> returns a value of type <code>int</code> . In SunOS release 5.4, ABI, SVID, and SVR4, <code>getegid()</code> returns a value of type <code>gid_t</code> . Also, SunOS release 5.4, 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
getegid(2V) — SysV	S		S	S	S	N
geteuid(2V)	C	In SunOS release 4.x, <code>geteuid()</code> returns a value of type <code>int</code> . In SunOS release 5.4, ABI, SVID, and SVR4, <code>geteuid()</code> returns a value of type <code>uid_t</code> . Also, SunOS release 5.4, 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
geteuid(2V) — SysV	S		S	S	S	N

≡ B

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Release 5.4	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 5.4, ABI, SVID, and SVR4, <code>getgid()</code> returns a value of type <code>gid_t</code> . Also, SunOS release 5.4, 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 5.4, ABI, SVID, and SVR4, the <code>grouplist</code> argument is of type <code>gid_t</code> . Also, SunOS release 5.4, 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	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	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	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 5.4 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 5.4	Notes	ABI	SVID	SVR4	BSD
getpgrp(2V)	C	The SunOS release 4.x version of <code>getpgrp()</code> has an argument <i>pid</i> , and <code>getpgrp()</code> returns the process group of the process indicated by <i>pid</i> . SunOS release 5.4, 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 5.4, ABI, SVID, and SVR4 <code>getpgrp()</code> return a value of type <code>pid_t</code> . SunOS release 5.4, 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 5.4, ABI, SVID, and SVR4, <code>getpid()</code> returns a value of type <code>pid_t</code> . Also, SunOS release 5.4, 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 5.4, ABI, SVID, and SVR4, <code>getppid()</code> returns a value of type <code>pid_t</code> . Also, SunOS release 5.4, 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	The <code>prctl(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 Release 5.4	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 5.4, ABI, SVID, and SVR4 versions. SunOS release 5.4, 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 5.4, 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)	N	The <code>sysconf(3C)</code> routine provides similar functionality. The <code>ru_utime</code> and <code>ru_stime</code> elements are of type <code>timeval</code> in SunOS release 4.x, in the <code>getrusage(3B)</code> routine, these are of type <code>timestruct_t</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 5.4 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 5.4, ABI, SVID, and SVR4, <code>getuid()</code> returns a value of type <code>uid_t</code> . Also, SunOS release 5.4, 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 165	C	C	C	N

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Release 5.4	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 5.4, 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 5.4, ABI, SVID, and SVR4, the <i>pid</i> argument is of type <code>pid_t</code> . Also, SunOS release 5.4, 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 5.4, 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 5.4, argument <i>len</i> is of type <code>size_t</code> which is defined to be <code>unsigned int</code> . SunOS release 5.4 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 5.4	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 5.4, ABI, SVID, and SVR4, the mode argument is of type <code>mode_t</code> . Also, SunOS release 5.4, 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 5.4, 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 5.4, 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 5.4, 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 5.4, 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 5.4	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 5.4, 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 5.4, 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 5.4, 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 5.4, ABI, SVID, and SVR4 <code><sys/mman.h></code> .	C	C	C	N

≡ B

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Release 5.4	Notes	ABI	SVID	SVR4	BSD
mount(2)	C#	<p>The SunOS release 4.x version of mount() and the SunOS 5.4 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 5.4, ABI, SVID, and SVR4, the first argument is <i>fs</i>, is the name of the file system. In SunOS release 5.4, 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 5.4, ABI, SVID, and SVR4 version uses two parameters (five and six: <code>const char *dataptr</code> and <code>int datalen</code>). Also, SunOS release 5.4, 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_MULTI) which are not defined in SunOS release 5.4, 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 5.4, ABI, SVID, or SVR4. The functionality of the following flags, defined in <code><sys/mount.h></code>, is not supported by the SunOS 5.4 release, or the ABI, SVID, or SVR4 versions: M_NOSUB, M_GRPID, M_MULTI.</p> <p>SunOS release 4.x mount() uses the following <code>errno</code> values which are not returned by the SunOS 5.4 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
msgrcv(2)	S		S	S	S	N

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Release 5.4	Notes	ABI	SVID	SVR4	BSD
msgsnd(2)	S		S	S	S	N
msync(2)	S		S#	S#	S	N
munmap(2)	S		S	S	S	N
nfssvc(2)	A	This interface is replaced in SunOS release 5.4 by the <code>nfssys(NFS_SVC, . . .)</code> routine.	N	N	N	N
open(2V)	C#	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 5.4, 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 5.4, 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 5.4, 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. Also, SunOS release 5.4, 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 <code>errno</code> value is valid for the SunOS 4.1 version of this system call but is not returned in SunOS release 5.4, ABI, SVID, and SVR4 versions: <code>EOPNOTSUPP</code> .	C#	C#	C#	N
open(2V) — SysV	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 5.4, ABI, SVID, and SVR4 versions: <code>EOPNOTSUPP</code> .	S#	S#	S#	N
pathconf(2V) — SysV	S		S	S	S	N
pipe(2V) — SysV	S		S	S	S	N
poll(2)	S		S	S	S	N
profil(2)	S		S	S	S	N

≡ B

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Release 5.4	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 5.4 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 5.4. 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 5.4. Also, the SunOS 5.4 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 5.4 version: <code>EPERM</code> . See “ <code>ptrace()</code> Request Values” on page 168 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>fcntl(2V)</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 5.4, 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 5.4. The SunOS 5.4 <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 5.4 version: <code>EISDIR</code> , <code>EWOULDBLOCK</code> .	C#	C#	C#	N
<code>readlink(2)</code>	S		S	S	S	N
<code>readv(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 5.4, ABI, SVID, and SVR4 versions: <code>EISDIR</code> , <code>EWOULDBLOCK</code> .	C#	C#	C#	N

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Release 5.4	Notes	ABI	SVID	SVR4	BSD
<code>readv(2V)</code> — SysV	C#	SunOS release 4.x and SunOS release 5.4, 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 5.4, or the SVID or SVR4 <code>iov_len</code> is defined as unsigned. SunOS release 5.4, 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
<code>reboot(2)</code>	A	The <code>uadmin(2)</code> system call provides similar functionality.	N	N	N	S
<code>recv(2)</code>	S	Now <code>recv(3N)</code> .	N	N	S	N
<code>recvfrom(2)</code>	S	Now <code>recvfrom(3N)</code> .	N	N	S	N
<code>recvmsg(2)</code>	S	Now <code>recvmsg(3N)</code> .	N	N	S	N
<code>rename(2V)</code> — SysV	S#	The SunOS release 5.4, 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 5.4, ABI, SVID, and SVR4 versions: <code>ENOTEMPTY</code> . SunOS release 5.4, ABI, SVID, and SVR4 versions set <code>errno</code> to flag <code>EEXIST</code> instead.	S#	S#	S#	N
<code>rmdir(2V)</code> — SysV	S#	The SunOS release 5.4, 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 5.4, ABI, SVID, and SVR4 versions: <code>ENOTEMPTY</code> . SunOS release 5.4, ABI, SVID, and SVR4 versions set <code>errno</code> to flag <code>EEXIST</code> instead.	S#	S#	S#	N
<code>sbrk(2)</code>	S		N	N	S	N
<code>select(2)</code>	S	Now <code>select(3C)</code> .	N	N	S	N
<code>semctl(2)</code>	S		S	S	S	N
<code>semget(2)</code>	S		S	S	S	N
<code>semop(2)</code>	S		S	S	S	N

≡ B

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Release 5.4	Notes	ABI	SVID	SVR4	BSD
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 5.4 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 5.4 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 5.4 version: <code>ENOBUFS</code> .	N	N	S#	N
setaudit(2)	N		N	N	N	N
setauid(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 5.4, ABI, SVID, and SVR4, the <code>grouplist</code> argument is of type <code>gid_t</code> . Also, SunOS release 5.4, 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	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 Release 5.4	Notes	ABI	SVID	SVR4	BSD
setpgrp(2V)	C#	The SunOS release 4.x version of <code>setpgrp()</code> has arguments <code>pid</code> and <code>pgrp</code> , and <code>setpgrp()</code> sets the process group to <code>pgrp</code> of the process indicated by <code>pid</code> . The SunOS 5.4 version of <code>setpgrp()</code> does not accept an argument, and <code>setpgrp()</code> also creates a new session. However, if <code>pgrp</code> is zero and <code>pid</code> refers to the calling process, then SunOS release 4.x <code>setpgrp()</code> call is identical to a SunOS 5.4 <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 5.4, <code>setpgrp()</code> returns a value of type <code>pid_t</code> . Also, the SunOS 5.4 version includes <code><unistd.h></code> and <code><sys/types.h></code> while the SunOS release 4.x version does not. In SunOS release 5.4, 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 5.4, 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 5.4, 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 5.4, ABI, SVID, and SVR4 versions: <code>EACCES</code> , <code>EINVAL</code> , <code>ESRCH</code> .	S#	S#	S#	N
setpriority(2)	A	The <code>pricntl(2)</code> system call provides similar functionality.	A	A	A	S
setregid(2)	A	The <code>setgid(rgid)</code> and <code>setegid(egid)</code> routines provide similar functionality in SunOS release 5.4.	N	N	N	C
setreuid(2)	A	The <code>setuid(ruid)</code> and <code>seteuid(euid)</code> routines provide similar functionality in SunOS release 5.4.	N	N	N	C
setrlimit(2)	C	See <code>getrlimit(2)</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

≡ B

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Release 5.4	Notes	ABI	SVID	SVR4	BSD
setuseraudit(2)	N		N	N	N	N
sgetl(2)	S	Now sgetl(3X).	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 release 2.4 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 5.4, 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 5.4 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 5.4, ABI, SVID, and SVR4.	A	A	A	S
sigstack(2)	A	The sigaltstack(2) system call provides similar functionality.	A	A	A	S

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Release 5.4	Notes	ABI	SVID	SVR4	BSD
sigsuspend(2V) — SysV	S		S	S	S	N
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 5.4 PF_IMPIPKNK is not supported. The following <code>errno</code> flags are valid for SunOS release 4.x <code>socket()</code> system call but are not valid in the SVR4 and SunOS 5.4 version: ENOBUFS, EPROTOTYPE.	N	N	C#	N
socketpair(2)	S	Now socketpair(3N).	N	N	S	N
sputl(2)	S	Now sputl(3X).	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 uadmin(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 <code>umask()</code> , but not by SunOS release 5.4, ABI, SVID, and SVR4 versions: S_IREAD (00400), S_IWRITE (00200), S_IEXEC (00100). However, the equivalent SunOS release 5.4, 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

≡ B

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Release 5.4	Notes	ABI	SVID	SVR4	BSD
umount(2V) — SysV	S		S	S	S	N
uname(2V) — SysV	S		S	S	S	N
unlink(2V) — SysV	S		S	S	S	N
umount(2)	A	The umount(2) 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, wait() returns a value of type int. In SunOS release 5.4, ABI, SVID, and SVR4, wait() returns a value of type pid_t. Also, SunOS release 5.4, ABI, SVID, and SVR4 versions include <sys/types.h> while the SunOS release 4.x version does not. The union wait, supported in SunOS release 4.x for backwards compatibility with previous SunOS releases, is not supported in SunOS release 5.4, ABI, SVID, and SVR4 versions. In SunOS release 4.x, wait() is automatically restarted when a process receives a signal while awaiting termination, unless the SV_INTERRUPT bit is set in the flags for that signal. In SunOS release 5.4, ABI, SVID, and SVR4, the wait() system call returns prematurely if a signal is received.	C	C	C	N
wait(2V) — SysV	C	The union wait, supported in SunOS release 4.x for backwards compatibility with previous SunOS releases, is not supported in SunOS release 5.4, ABI, SVID, and SVR4 versions. In SunOS release 4.x, wait(2V) is automatically restarted when a process receives a signal while awaiting termination, unless the SV_INTERRUPT bit is set in the flags for that signal. In SunOS release 5.4, ABI, SVID, and SVR4, the wait(2) function will return prematurely if a signal is received.	C	C	C	N
wait3(2V)	A	The wait(2) and waitpid(2) system calls provide similar functionality.	A	A	A	S

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Release 5.4	Notes	ABI	SVID	SVR4	BSD
wait4(2V)	A	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 5.4, ABI, SVID, and SVR4, waitpid() returns a value of type <code>pid_t</code> . The <i>pid</i> argument to waitpid() is of type <code>int</code> in SunOS release 4.x and of type <code>pid_t</code> in SunOS release 5.4, ABI, SVID, and SVR4. Also, SunOS release 5.4, 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 5.4, 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 5.4, 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 5.4, 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 5.4, ABI, SVID, and SVR4, the waitpid(2) function will return prematurely if a signal is received.	C	C	C	N

≡ B

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Release 5.4	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 FIONBIO request to <code>ioctl()</code>, or by using <code>fcntl()</code> to set the FNDELAY or O_NDELAY flag, <code>write()</code> returns -1 and sets <code>errno</code> to EWOULDBLOCK.</p> <p>In SunOS release 5.4, on a <code>write()</code> to a regular file, if O_NDELAY or O_NONBLOCK is set, <code>write()</code> returns -1 and sets <code>errno</code> to EAGAIN.</p> <p>On <code>write()</code> requests to a pipe or FIFO with O_NONBLOCK or O_NDELAY 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 O_NONBLOCK is set, it returns -1 and sets <code>errno</code> to EAGAIN and when O_NDELAY is set, it returns 0.</p> <p>With O_NDELAY set, <code>write()</code> requests for {PIPE_BUF} or fewer bytes either succeed completely and return <i>nbytes</i>, or return 0. A <code>write()</code> request for greater than {PIPE_BUF} bytes either transfers what it can and returns the number of bytes written, or transfers no data and returns 0. Also, if a request is greater than {PIPE_BUF} bytes and all data previously written to the pipe has been read, <code>write()</code> transfers at least {PIPE_BUF} bytes.</p> <p>The SunOS 5.4 <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 5.4 version: EWOULDBLOCK.</p>	C#	C#	C#	N
write(2V) — SysV	C#	<p>The SunOS release 5.4, 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 5.4, ABI, SVID, and SVR4 versions: EWOULDBLOCK.</p>	C#	C#	C#	N

Table B-1 System Calls Reference Table (Continued)

SunOS Release 4.x System Call	SunOS Release 5.4	Notes	ABI	SVID	SVR4	BSD
writev(2V)	C#	SunOS release 5.4, ABI, SVID, and SVR4 versions of <code>writev()</code> 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 5.4 release, or the ABI, SVID, or SVR4 version: <code>EWOULDBLOCK</code> .	C#	C#	C#	N

≡ B

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 2.4, 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 2.4 command is provided.
- If an interface is listed as “the same” (S), the Solaris 2.4 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 5.4 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 2.4 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 5.4	Alternative Available and Notes	ABI	SVID	SVR4	BSD
<code>bcopy(3)</code>	<code>memcpy()</code>	This routine is replaced by <code>memcpy(3C)</code> in SunOS release 5.4, or the ABI, SVID, or SVR4.	A	A	A	S

The `bcopy()` routine has been replaced by `memcpy()` in SunOS release 5.4. Applications that need to run on ABI, SVID or SVR4 compliant systems should also use `memcpy()`. A version of `bcopy()`, which is similar to the SunOS version, is available with the SunOS/BSD Compatibility package. Applications that use this version will not be compatible with other SVR4 systems.

SunOS release 4.x Command	SunOS release 5.4	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 5.4 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 5.4	Alternative Available and Notes	ABI	SVID	SVR4	BSD
putpwent(3)	S		S	S	S	N

The SunOS release 4.x `putpwent()` routine and the SunOS release 5.4 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 5.4	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 5.4, 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 5.4, 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 5.4	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 5.4, 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 5.4. 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 5.4	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 5.4, 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 5.4, 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 5.4, 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
attroff(3V) -- SysV	S		N	S	S	N
attron(3V) -- SysV	S		N	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	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 superseded by authdes_seccreate(3N) in SunOS release 5.4, 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 superseded by authsys_seccreate(3N) in SunOS release 5.4, or the ABI, SVID, or SVR4.	A	A	A	N
authunix_create_default(3N)	authsys _create_default()	This routine is still available, but is superseded by authsys_create_default(3N) in SunOS release 5.4, or the ABI, SVID, or SVR4.	A	A	A	N
baudrate(3V) -- SysV	S		N	S	S	N
bcmp(3)	memcmp()	This routine is replaced by memcmp(3C) in SunOS release 5.4, or the ABI, SVID, or SVR4.	A	A	A	S
bcopy(3)	memcpy()	This routine is replaced by memcpy(3C) in SunOS release 5.4, or the ABI, SVID, or SVR4.	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 5.4	Notes	ABI	SVID	SVR4	BSD
box(3V)	C	The SunOS 4.x version of <code>box()</code> is a function while the SunOS release 5.4, 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 5.4, 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 5.4, 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)	A	This routine is replaced by <code>memset(3C)</code> in SunOS release 5.4, or the ABI, SVID, or SVR4.	A	A	A	S
calloc(3)	S		S	S	S	N
callrpc(3N)	<code>rpc_call()</code>	This routine is still available, but is superseded by <code>rpc_call(3N)</code> in SunOS release 5.4, or the ABI, SVID, or SVR4.	N	N	S	N
catclose(3C)	S		S	S	S	N
catgetmsg(3C)	<code>catgets()</code> and <code>strncpy()</code>	In SunOS release 5.4, 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

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes	ABI	SVID	SVR4	BSD
catopen(3C)	S		S	S	S	N
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)	free()	This routine is replaced by void free(void*ptr) (refer to malloc(3C)) in SunOS release 5.4, 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)	rpc _broadcast()	This routine is still available, but is superseded by rpc_broadcast(3N) in SunOS release 5.4, 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 5.4	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 5.4 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 5.4, 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 5.4, 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 5.4, 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 5.4	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 5.4, 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 5.4, 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 5.4, 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 5.4	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 5.4, or the ABI, SVID, or SVR4 <code>tm</code> structure: <code>tm_zone</code> and <code>tm_gmtoff</code>. Instead SunOS release 5.4, 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 5.4, 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 5.4, 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 5.4, 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 5.4	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 5.4	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 5.4	Notes	ABI	SVID	SVR4	BSD
endexportent(3)	A	The <code>/etc/dfs/sharetab</code> file replaces <code>/etc/exports</code> in SunOS release 5.4, 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 5.4, 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 5.4, 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 5.4 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 5.4, 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 5.4	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	SunOS release 5.4, or the ABI, SVID, or SVR4 version of <code>execl()</code> includes <code><unistd.h></code> while the SunOS 4.x version does not.	C	C	C	N
execl(3V) -- SysV	S		S	S	S	N
execle(3V)	C	SunOS release 5.4, or the ABI, SVID, or SVR4 version of <code>execle()</code> includes <code><unistd.h></code> while the SunOS 4.x version does not.	C	C	C	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes	ABI	SVID	SVR4	BSD
<code>execle(3V)</code> -- SysV	S		S	S	S	N
<code>execlp(3V)</code>	C	SunOS release 5.4, or the ABI, SVID, or SVR4 version of <code>execlp()</code> includes <code><unistd.h></code> while the SunOS 4.x version does not.	C	C	C	N
<code>execlp(3V)</code> -- SysV	S		S	S	S	N
<code>execv(3V)</code>	C	SunOS release 5.4, or the ABI, SVID, or SVR4 version of <code>execv()</code> includes <code><unistd.h></code> while the SunOS 4.x version does not.	C	C	C	N
<code>execv(3V)</code> -- SysV	S		S	S	S	N
<code>execvp(3V)</code>	C	SunOS release 5.4, or the ABI, SVID, or SVR4 version of <code>execvp()</code> includes <code><unistd.h></code> while the SunOS 4.x version does not.	C	C	C	N
<code>execvp(3V)</code> -- SysV	S		S	S	S	N
<code>exit(3)</code>	C	Both the SunOS 4.x and SunOS release 5.4, 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 5.4, 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 5.4, or the ABI, SVID, or SVR4 versions of <code>exit()</code> are compatible.	C	C	C	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes	ABI	SVID	SVR4	BSD
exp(3M)	C	In SunOS release 5.4, 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 5.4, 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
exportent(3)	A	The <code>/etc/dfs/sharetab</code> file replaces <code>/etc/exports</code> in SunOS release 5.4, 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
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)	<code>dbm_fetch()</code>	This routine is replaced by <code>dbm_fetch(3)</code> in SunOS release 5.4.	N	N	N	S
fflush(3S)	S		S	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes	ABI	SVID	SVR4	BSD
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
firstkey(3X)	dbm _firstkey()	This routine is replaced by dbm_firstkey(3) in SunOS release 5.4.	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 5.4, 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

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes	ABI	SVID	SVR4	BSD
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)	time()	This routine is replaced by time(2) in SunOS release 5.4, or the ABI, SVID, or SVR4.	A	A	A	S
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

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes	ABI	SVID	SVR4	BSD
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 5.4, 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
getcwd(3V)	S	The SVR4 and SunOS 5.4 getcwd() routine is compatible with the SunOS 4.x getcwd(). In the SunOS release 4.x, if buf is a NULL pointer, getcwd() obtains size bytes of space using malloc(3). This capability is not supported by the ABI and SVID version of getcwd().	C	C	S	N
getenv(3V)	S		S	S	S	N
getexportent(3)	A	The /etc/dfs/sharetab file replaces /etc/exports in SunOS release 5.4, or the ABI, SVID, or SVR4. Refer to share(1M), unshare(1M), and sharetab(4) for more information.	A	A	A	N
getexportopt(3)	A	The /etc/dfs/sharetab file replaces /etc/exports in SunOS release 5.4, or the ABI, SVID, or SVR4. Refer to share(1M), unshare(1M), and sharetab(4) for more information.	A	A	A	N
getfauditflags(3)	N		N	N	N	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes	ABI	SVID	SVR4	BSD
getfsent(3)	getvfsent()	This routine is replaced by <code>getvfsent(3)</code> in SunOS release 5.4.	N	N	N	N
getfsfile(3)	getvfsfile()	This routine is replaced by <code>getvfsfile(3)</code> in SunOS release 5.4.	N	N	N	N
getfsspec(3)	getvfsfile()	This routine is replaced by <code>getvfsfile(3)</code> in SunOS release 5.4.	N	N	N	N
getfstype(3)	getvfsany()	This routine is replaced by <code>getvfsany(3)</code> in SunOS release 5.4.	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
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 5.4, 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 5.4, or the SVID or SVR4 <code>getmntent()</code> returns <code>int</code> . Additionally, SunOS release 5.4, 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

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes	ABI	SVID	SVR4	BSD
get_myaddress(3N)	S	This routine is still available, but is superseded by <code>netdir_getbyname(3N)</code> in SunOS release 5.4, 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
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)	N	<code>pmap_getport()</code> can be used to get the same result.	N	N	N	N
gets(3S)	S		S	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes	ABI	SVID	SVR4	BSD
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 5.4 header file <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	N	N	S
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 5.4 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 5.4 tty system.	N	N	N	N
getusershell(3)	S		N	N	N	N
getw(3V)	S		S	S	S	N
getwd(3)	getcwd()	This routine is replaced by <code>char *getcwd(char *buf, int size)</code> in SunOS release 5.4, or the ABI, SVID, or SVR4.	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

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes	ABI	SVID	SVR4	BSD
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
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 5.4, 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)	<code>strchr()</code>	This routine is replaced by <code>strchr(3C)</code> in SunOS release 5.4, or the ABI, SVID, or SVR4.	A	A	A	S

≡ C

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes	ABI	SVID	SVR4	BSD
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 5.4, 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 ERR, while SunOS release 5.4, or the SVID or SVR4 writes an appropriate error message to the standard error and exits.	N	C	C	S
initscr(3V) -- SysV	S		N	S	S	N
initstate(3)	drand48()	The <code>drand48(3N)</code> routines replace this interface in SunOS release 5.4, or the SVID or SVR4.	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

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes	ABI	SVID	SVR4	BSD
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)	N		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
isupper(3V)	S	In the SunOS release 4.x, <code>j0(HUGE)</code> , <code>j1(HUGE)</code> , <code>jn(4,HUGE)</code> will return zero with no error indication. In SunOS release 5.4, or the SVID or SVR4 these routines will return zero, set <code>errno</code> to <code>ERANGE</code> , 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

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes	ABI	SVID	SVR4	BSD
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
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)	N		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

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes	ABI	SVID	SVR4	BSD
ldclose(3X)	N		N	N	N	N
ldexp(3M)	C	SunOS 4.x version of <code>ldexp()</code> differs from SunOS release 5.4, 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 5.4, 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
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

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes	ABI	SVID	SVR4	BSD
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 5.4, or the SVID or SVR4, it returns -HUGE with an EDOM error and DOMAIN matherr.	N	C	C	N
log10(3M)	C	In the SunOS release 4.x, when log10() produces undefined results (for example, log10(0)) it returns NaN, with an EDOM error and a DOMAIN matherr. In SunOS release 5.4, 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

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes	ABI	SVID	SVR4	BSD
longname(3V)	C	The SunOS 4.x version of longname() requires two arguments, <i>termbuf</i> and <i>name</i> , which do not exist in SunOS release 5.4, or the SVID or SVR4 version. <i>termbuf</i> is a pointer to the terminal entry from termcap, which is replaced by terminfo in SunOS release 5.4, 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 5.4, 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
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

≡ C

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes	ABI	SVID	SVR4	BSD
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_stkcswwset(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
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

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes	ABI	SVID	SVR4	BSD
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
mktemp(3)	C	The SunOS 4.x <code>mktemp()</code> routine replaces the trailing X characters of template with a letter and the current process ID. SunOS release 5.4, 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 5.4, or the ABI, SVID, or SVR4 versions of <code>mktemp()</code> are compatible.	C	C	C	N
mlock(3)	S		S	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes	ABI	SVID	SVR4	BSD
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)	profil()	This routine is replaced by <code>profil(2)</code> in SunOS release 5.4, or the ABI, SVID, or SVR4.	A	A	A	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes	ABI	SVID	SVR4	BSD
monitor(3)	C	<p>The SunOS 4.x <code>monitor()</code> routine differs from SunOS release 5.4, 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>externetext(); monitor(N_TXTOFF(0),etext, buf, bufsize, nfunc);</pre> <p>While with SunOS release 5.4, or the ABI, SVID, or SVR4 <code>monitor()</code> routine, it is sufficient to use:</p> <pre>externintetext(); 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 5.4, 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 5.4, 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 5.4	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 5.4, 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 5.4, 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 5.4, 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 5.4, 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 5.4	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 5.4, 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 5.4, 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 5.4	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 5.4, 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 5.4, or the SVID or SVR4 version returns 0 on success, and -1 on error. Note that SunOS release 5.4 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 5.4	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 5.4, 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 5.4, 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 5.4, 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 5.4, or the ABI, SVID, or SVR4 version includes <code><sys/types.h></code> while the SunOS 4.x version does not. SunOS release 5.4, 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 5.4	Notes	ABI	SVID	SVR4	BSD
overlay(3V)	C	The SunOS 4.x <code>overlay()</code> is a function while SunOS release 5.4, 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 5.4, 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 5.4, 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 5.4, 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 5.4, or the ABI, SVID, or SVR4.	A	A	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes	ABI	SVID	SVR4	BSD
pmap_getport(3N)	S	This routine is still available, but is superseded by <code>rpcb_getaddr(3N)</code> in SunOS release 5.4, or the ABI, SVID, or SVR4.	A	A	S	N
pmap_rmtcall(3N)	S	This routine is still available, but is superseded by <code>rpcb_rmtcall(3N)</code> in SunOS release 5.4, or the ABI, SVID, or SVR4.	A	A	S	N
pmap_set(3N)	S	This routine is still available, but is superseded by <code>rpcb_set(3N)</code> in SunOS release 5.4, or the ABI, SVID, or SVR4.	A	A	S	N
pmap_unset(3N)	S	This routine is still available, but is superseded by <code>rpcb_unset(3N)</code> in SunOS release 5.4, or the ABI, SVID, or SVR4.	A	A	S	N
pnp(3R)	N		N	N	N	N
pnoutrefresh(3V) -- SysV	S		N	S	S	N
pod_exit(3L)	N		N	N	N	N
pod_getexit(3L)	N		N	N	N	N
pod_getmaxpri(3L)	N		N	N	N	N
pod_getmaxsize(3L)	N		N	N	N	N
pod_setexit(3L)	N		N	N	N	N
pod_setmaxpri(3L)	N		N	N	N	N
point(3X)	S		N	N	N	N
popen(3S)	S		S	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes	ABI	SVID	SVR4	BSD
<code>pow(3M)</code>	C	In SunOS release 5.4, 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 5.4, 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 5.4, 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 5.4, 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 <code>void</code> . SunOS release 5.4, 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 5.4	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 5.4.	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 5.4, 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 5.4	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 5.4, or the ABI, SVID or SVR4. Applications must migrate to the <code>dirent</code> structure defined in <code><dirent.h></code> . SunOS release 5.4, 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 5.4 only have the <code>d_name</code> and <code>d_reclen</code> fields in common. Also, SunOS release 5.4 <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 5.4 only have the <code>d_name</code> and <code>d_reclen</code> fields in common. Also, SunOS release 5.4 <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 5.4, 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 5.4, 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 5.4	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 5.4.	A	A	A	S
re_exec(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>regex(3G)</code> in SunOS release 5.4.	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 5.4, 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 5.4, 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 5.4, 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 5.4	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)	N		A	A	A	S
rint(3M)	strchr()	This routine is replaced by <code>strchr(3C)</code> in SunOS release 5.4, or the ABI, SVID, or SVR4.	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 <code>def_prog_mode(3X)</code> in SunOS release 5.4, 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 5.4	Notes	ABI	SVID	SVR4	BSD
scalb(3M)	C	In SunOS release 5.4, 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 5.4, 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 5.4	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 5.4, 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)	N		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 5.4. Refer to share(1M), unshare(1M), and sharetab(4) for more information.	N	N	N	N
setfsent(3)	fopen()	This routine is replaced by fopen(3) in SunOS release 5.4, 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 5.4	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)	N		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)	setgid()	This routine is replaced by setgid(2) in SunOS release 5.4, or the ABI, SVID, or SVR4.	A	A	A	N
setrpcent(3N)	S		N	N	S	N
setruid(3V)	setuid()	This routine is replaced by setuid(2) in SunOS release 5.4, or the ABI, SVID, or SVR4.	A	A	A	N
setscreg(3V) -- SysV	S		N	S	S	N
setservent(3N)	S		N	N	S	N
setstate(3)	drand48()	The drand48(3N) routines replace this interface in SunOS release 5.4, or the SVID or SVR4.	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 5.4	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 5.4. 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 5.4 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)	N		N	N	N	N
siginterrupt(3V)	<code>sigaction()</code>	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 5.4	Notes	ABI	SVID	SVR4	BSD
signal(3V)	C	The following SunOS 4.x signal is not defined in the SVR4 and SunOS 5.4 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 5.4, 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 5.4	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 5.4, 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 5.4, 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)	<code>rand48()</code> and <code>rand()</code>	The <code>rand48(3C)</code> (for SunOS release 5.4, 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
signal(3)	S		N	N	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	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 5.4.	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 5.4	Notes	ABI	SVID	SVR4	BSD
strftime(3V)	C	<p>There are some differences in the directives specified in the format:</p> <ul style="list-style-type: none"> %k and %l - Not supported in SunOS release 5.4. %S- SunOS 4.x specifies seconds to be in the range of 0-59, while SunOS release 5.4 defines seconds to be in the range of 0-61 (allows for lead seconds). %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 5.4, week number 01 is the first week in January starting with a Sunday for %U or a Monday for %W. <p>The SunOS4.1 <code>tmstructure</code> contains two fields not present in SunOS release 5.4 <code>tmstructure</code>: <code>tm_zone</code> and <code>tm_gmtoff</code>. Instead SunOS release 5.4 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 5.4 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 5.4	Notes	ABI	SVID	SVR4	BSD
strpbrk(3)	S		S	S	S	N
strptime(3V)	getdate()	This routine is replaced by <code>getdate(3C)</code> in SunOS release 5.4, 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 5.4, 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 5.4, or the SVID or SVR4 routine returns a null pointer if failure. SunOS release 5.4, 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 5.4, the SVID or SVR4.	N	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	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 superseded by <i>svc_getrpccaller(3N)</i> in SunOS release 5.4, or the ABI, SVID, or SVR4.	A	A	A	N
svc_getreq(3N)	S	This routine is still available, but is superseded by <i>svc_getreqset(3N)</i> in SunOS release 5.4, or the ABI, SVID, or SVR4.	S	S	S	N
svc_getreqset(3N)	S		S	S	S	N
svc_register(3N)	<i>svc_reg()</i>	This routine is still available, but it is superseded by <i>svc_reg(3N)</i> in SunOS release 5.4, 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)	<i>svc_unreg()</i>	This routine is still available, but is superseded by <i>svc_unreg(3N)</i> in SunOS release 5.4, 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 5.4	Notes	ABI	SVID	SVR4	BSD
svdfd_create(3N)	svc_fd_create()	This routine is still available, but is superseded by <code>svc_fd_create(3N)</code> in SunOS release 5.4, 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 5.4, 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 5.4, 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 5.4, 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 5.4, or the ABI, SVID, or SVR4.	N	N	S	S
swab(3)	S		S	S	S	N
sys_siglist(3)	N		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

≡ C

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	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 5.4	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 5.4, or the SVID or SVR4 is supporting this routine as a conversion aid and it should not be used in new applications. SunOS release 5.4, 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 5.4, or the SVID or SVR4 is supporting this routine as a conversion aid and it should not be used in new applications. SunOS release 5.4, 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 5.4, or the SVID or SVR4 is supporting this routine as a conversion aid and it should not be used in new applications. SunOS release 5.4, 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 5.4	Notes	ABI	SVID	SVR4	BSD
tgetstr(3X)	C	SunOS release 5.4, or the SVID or SVR4 is supporting this routine as a conversion aid and it should not be used in new applications. SunOS release 5.4, 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 5.4, or the SVID or SVR4 is supporting this routine as a conversion aid and it should not be used in new applications. SunOS release 5.4, 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 5.4, or the ABI, SVID, or SVR4.	A	A	A	N
timelocal(3V)	S	This routine is replaced by localtime(3C) in SunOS release 5.4, 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 5.4, 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 5.4	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 5.4, 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 5.4, 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 5.4, 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 5.4, 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 5.4, or the SVID or SVR4 is supporting this routine as a conversion aid and it should not be used in new applications. SunOS release 5.4, 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 5.4	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
ttynam(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)	tzset()	This routine is replaced by <code>tzset(3C)</code> in SunOS release 5.4, or the ABI, SVID, or SVR4.	A	A	A	N
ualarm(3)	setitimer()	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 5.4 <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 5.4	Notes	ABI	SVID	SVR4	BSD
<code>ungetc(3S)</code>	S	The SVR4 and SunOS 5.4 <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)</code> -- SysV	S		N	S	S	N
<code>user2netname(3N)</code>	S		S	S	S	N
<code>usleep(3)</code>	<code>setitimer()</code> and <code>select()</code>	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 5.4, 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 5.4, 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)</code> -- SysV	S		N	S	S	N
<code>vidputs(3V)</code> -- SysV	S		N	S	S	N
<code>vlimit(3C)</code>	<code>getrlimit()</code>	This routine is replaced by <code>getrlimit(2)</code> in SunOS release 5.4, or the ABI, SVID, or SVR4.	A	A	A	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes	ABI	SVID	SVR4	BSD
vprintf(3V)	C	The SunOS 4.x vprintf(), vfprintf(), and vsprintf() routines are incompatible with SunOS release 5.4, 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 5.4, 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 5.4, or the ABI, SVID, or SVR4.	N	N	N	N
vtimes(3C)	getrusage()	This routine is replaced by getrusage(2) in SunOS release 5.4.	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 5.4 are chtype (long) if CHTYPE is not defined differently for <curses.h>. SunOS release 5.4, or the SVID or SVR4 version returns ERR on failure and an integer value other than ERR upon successful completion. SunOS release 5.4, 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 5.4	Notes	ABI	SVID	SVR4	BSD
waddch(3V) -- SysV	S		N	S	S	N
waddstr(3V)	C	SunOS release 5.4, or the SVID or SVR4 version of waddstr(3V) returns ERR (-1) on failure. SunOS release 5.4, 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 5.4, or the SVID or SVR4 version of wclear() routine always returns (OK = 0) upon success while SunOS 4.x returns void. SunOS release 5.4, 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 5.4, or the SVID or SVR4 version of wclrtoobot(3V) routine always returns (OK = 0) upon success while SunOS 4.x returns void. SunOS release 5.4, 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 5.4	Notes	ABI	SVID	SVR4	BSD
wclrtoeol(3V) -- SysV	S		N	S	S	N
wclrtoeol(3V)	C	SunOS release 5.4, or the SVID or SVR4 routine always returns (OK = 0) upon success while SunOS 4.x returns void. SunOS release 5.4, 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 5.4, or the SVID or SVR4 this routine may be a macro while it always is in SunOS 4.x. SunOS release 5.4, 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 5.4, or the SVID or SVR4 this routine may be a macro while it always is in SunOS 4.x. SunOS release 5.4, 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 5.4	Notes	ABI	SVID	SVR4	BSD
werase(3V)	C	In SunOS release 5.4, or the SVID or SVR4 this routine returns OK (0) or a non-negative integer if <code>immedok</code> is set. SunOS release 5.4, 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 5.4, 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 5.4, or the SVID or SVR4 version also has additional support for function keys.	N	C	C	S
wgetch(3V) -- SysV	C	In SunOS release 5.4, 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 5.4, 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 5.4, 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 5.4	Notes	ABI	SVID	SVR4	BSD
winsertln(3V)	C	SunOS release 5.4, or the SVID or SVR4 version of winsertln() returns ERR on failure and an integer value other than ERR upon successful completion. This can be a macro in SunOS release 5.4, or the SVID or SVR4.	N	C	C	S
winsertln(3V) -- SysV	S		N	S	S	N
wmove(3V)	C	SunOS release 5.4, or the SVID or SVR4 version of wmove() 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 5.4, or the SVID or SVR4 version of wprintw() returns ERR on failure and an integer value other than ERR upon successful completion. The SunOS 4.x version returns void. SunOS release 5.4, 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
wprintw(3V) -- SysV	S		N	S	S	N

Table C-1 Library Routines Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes	ABI	SVID	SVR4	BSD
wrefresh(3V)	C	SunOS release 5.4, 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 5.4, 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 5.4, 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 5.4, 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 5.4, 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 5.4	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 5.4, 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 5.4, 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 5.4	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 5.4	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 5.4, 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 5.4, 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 5.4, 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 2.4 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 2.4 command is provided.
- If an interface is listed as “the same” (S), the Solaris 2.4 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 2.4 interfaces, see the *man Pages(4): File Formats*.

≡ D

System Files

Table D-1 File Formats Reference Table

SunOS release 4.x	SunOS release 5.4	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
auto.vol(5)	S	Automount map for volumes
bar(5)	N	Tape archive file format
boards.pc(5)	N	ATN and XTN compatible boards for DOS windows
bootparams(5)	N	Boot parameter data base
bootservers(5)	N	NIS bootservers file
coff(5)	N	Common assembler and link editor output
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
ext_ports(5)	N	External ports file for network printers, terminals, and modems
fbtab(5)	N	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 5.4	Notes
fspec(5)	S	Format specification in text files
fstab(5)	A	Static filesystem mounting table, mounted filesystems table
ftpusers(5)	S	List of users prohibited by FTP
gettytab(5)	N	Terminal configuration data base
group(5)	S	Group file
group.adjunct(5)	N	Group security data file
help(5)	N	Help file format
help_viewer(5)	N	Help viewer file format
holidays(5)	C	Prime/non-prime table for System V accounting
hosts(5)	S	hHost name data base
hosts.equiv(5)	S	Trusted hosts by system and by user
indent.pro(5)	N	Default options for indent
inetd.conf(5)	S	Internet servers database
internat(5)	N	Key mapping table for internationalization
ipalloc.netrange(5)	N	Range of addresses to allocate
keytables(5)	S	Keyboard table descriptions for loadkeys and dumpkeys
link(5)	N	Link editor interfaces
locale(5)		Locale database
magic(5)	S	File command's magic number file
mtab(5)	A	Mounted file system table
netgroup(5)	S	List of network groups
netmasks(5)	S	Network mask data base
netrc(5)	S	File for ftp remote login data
networks(5)	S	Network name data base
orgrc(5)	N	Organizer configuration and initialization file
passwd(5)	C	Password file
passwd.adjunct(5)	N	User security data file

≡ D

Table D-1 File Formats Reference Table (Continued)

SunOS release 4.x	SunOS release 5.4	Notes
phones(5)	N	Remote host phone number data base
plot(5)	N	Graphics interface
pnpsysnames(5)	N	File used to allocate system names
policies(5)	N	Network administration policies
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)	N	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
setup.pc(5)	N	Master configuration file for DOS
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 5.4	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
ugid_alloc.range(5)	N	Range of user IDs and group IDs to allocate
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
vgrindefs(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

≡ *D*

/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 5.4 / 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/sunwsp	Mount/installation point for unbundled language products

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 2.4	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 2.4	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 2.4	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 2.4	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 2.4 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 5.4 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 2.4 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 2.4 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 2.4 kernel. Two systems have the same kernel architecture if the same Solaris 2.4 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 5.4 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 5.4 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 2.4 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 5.4 software distribution: for example, the SunPro compilers.

Index

A

ac, 208
accounting log file, 137
acctcms, 208
acctcom, 208
acctcon1, 208
acctcon2, 208
acctdisk, 208
acctdusg, 208
acctmerg, 208
accton, 208
acctprc1, 208
acctprc2, 208
acctwtmp, 208
adb, 162, 208
adbggen, 208
add_client, 208
add_services, 208
addbib, 208
adding devices, 67, 69
adding user accounts, 55
adjacentscreens, 208
admin, 209
administering port monitors, 130
administration commands, printer, 125
Administration Tool, 8
 and naming services, 20
 Database Manager, 20
 Host Manager, 20
 Printer Manager, 20, 22, 121, 122
 security, 47
 Serial Port Manager, 20
 User Account Manager, 20, 56
 using, 20
admintool, 20
adv, 209
advantages of migrating, 3
aedplot, 209
align_equals, 209
analyze, 209
application binary interface (ABI), 5
applications
 packaging, 174
 porting, 42
 SunOS release 4.x, 43
 well-behaved, 174
apropos, 209
ar, 209
arch, 209
architecture, definition, 363
arp, 209
as, 209

ASET (automated security enhancement tool), 48
at, 210
atoplot, 210
atq, 210
atrm, 210
audit, 210
audit_warn, 210
auditd, 210
auto configuration, 8
autofs, 17, 99
automated security enhancement tool (ASET), 48
automatic device driver loading, 9
automount, 210
automounter, 17, 18, 99
awk, 210

B

backing up files, 27, 32, 105
Backup Copilot, 28
banner, 210
bar, 105, 211
basename, 211
batch, 211
bc, 211
BFS file system, 88
bgplot, 211
biff, 211
/bin/csh, 51
/bin/sh, 51
binary compatibility mode, 34
binary compatibility package, 42
binary compatibility package, definition, 363
binary executable files, 34
bin-mail, 211
biod, 211
block, 107
boot, 74, 189, 211

boot block, 189
boot process, 73, 75
boot -r, using to add devices, 69
booting from PROM, 74
bootparamd, 211
Bourne shell, 51, 52
BSD/Source Compatibility Package, 42
building executables, 156
building shared libraries, 156

C

C language compatibility notes, 147 to 154
C shell, 51, 52
C2conv, 208
C2unconv, 208
CACHEFS pseudo file system, 82
CACHEFSpseudo file system, 87
cal, 211
calendar, 211
cancel, 211
capitalize, 211
captain, 211
cat, 211, 212
catman, 212
cb, 212
cc, 206, 212
cc, C compiler, 148 to 154
cd, 212
cdc, 212
/cdrom, 101
cflow, 212
change_login, 212
changing run levels, 76
character support, internationalization, 183
chargefee, 212
check4, 212
checkeq, 212
checknr, 212
checkpoint restart, 136

chfn, 212
 chgrp, 212
 chkey, 212
 chmod, 213
 chown, 213
 chroot, 213
 chrtbl, 213
 chsh, 213
 ckpacct, 213
 clear, 213
 clear_colormap, 213
 clear_function, 213
 click, 213
 client, 213
 client, definition, 363
 clients, SunOS release 4.x, 112
 clock, 213
 clri, 88, 213
 cluster, 12
 core system support cluster, 12
 developer system support cluster, 12
 end user system support cluster, 12
 entire distribution cluster, 12
 cluster, 213
 cluster, definition, 363
 cmdtool, 213
 cmp, 213
 col, 213
 colcrt, 214
 colldef, 214
 colldef(8), 184
 colltbl, 184
 coloredit, 214
 colrm, 214
 comb, 214
 comm, 214
 command locations, 90
 command log file, 137
 commands
 package commands, 13
 printer, 125
 SunOS 5.x
 SunOS release 4.x, 41
 commands reference table, 205 to 247
 commands, *See also* Appendix A
 compatibility, 5
 compatibility package, 67
 device names, 65
 compatibility packages, 41 to 43
 and NIS, 143
 and PATH, 42
 compilers, 147 to 154
 compress, 214
 concurrent processes, 136
 config, 188, 214
 configuration information, saving, 32
 configuration, definition, 363
 consistency checks, 105
 convert4x, 112, 119
 converting data, 36
 converting files, 37
 for NIS clients, 36
 for NIS+ master servers, 36
 with no name service, 36
 converting initialization files, 53
 copy_home, 214
 core system support cluster, 12
 cp, 214
 cpio, 105, 107, 214
 cpp, 214
 crash, 214
 cron, 214
 crontab, 214
 crtplot, 214
 crypt, 214
 csh, 214
 .cshrc, 34, 53
 csplit, 214
 ctags, 214
 ctrace, 215
 cu, 215
 customizing user environments, 53

cut, 215
cxref, 215
cylinder groups, 14

D

data, converting, 36
Database Manager, 20
 security, 47
dataless, definition, 363
date, 207, 215
dbconfig, 215
dbx, 162, 215
dbxtool, 162, 215
dc, 215
dcheck, 215
dd, 105, 107, 215
DDI (Device Driver Interface), 196, 197
DDI, definition, 363
DDI/DKI, 9
debugging object files, dump, 159
default directories, 83, 85
default dump device, 87
default file systems, 83, 85
default system access, 47
defaults_from_input, 216
defaults_merge, 216
defaults_to_indentpro, 216
defaults_to_mailrc, 216
defaultsedit, 216
delta, 216
deroff, 216
des, 216
DeskSet applications, 6
/dev, 15, 82, 91, 353
/dev/dsk, 91, 353
/dev/pts, 91, 353
/dev/rdsk, 91, 353
/dev/rmt, 91, 353
/dev/sad, 91, 353
/dev/term, 91, 353
developer system support cluster, 12
Device Driver Interface (DDI), 196, 197
device driver interfaces, 9
device driver loading, 9
device driver transition, 195
device drivers
 binary compatibility, 187
device information, 7
device naming, 65
 and SCSI disks, 66
 comparison of conventions, 67
 convention for disk drives, 66
 conventions for tape drives, 67
 SunOS release 4.1 conventions, 65
 SunOS release 5.4 conventions, 14
devices, adding, 67, 69
devinfo, 67, 69, 198, 216
devnm, 216
df, 67, 68, 88, 102, 216
diff, 217
diff3, 217
differences between OSs, 6
diffmk, 217
dircmp, 217
directories, default, 85
directory search path
 libraries, 157
 modules, 188
dirname, 217
dis, 217
disablenumlock, 217
discover4x, 112, 113 to ??, 113, ?? to 115
disk commands, 67
disk configuration information, saving, 28
disk drives
 and device naming, 66
 getting information about, 67
 label information, 68
disk names, 193
disk partition information, obtaining, 28
disk partition slice numbers, 66
disk partition, definition, 364

disk slice, 13
disk slice, definition, 364
diskettes, managing, 70
diskless, definition, 364
diskusg, 217
Distributed File System (DFS), 134
dkctl, 217
DKI (Driver Kernel Interface), 9, 196
DKI, definition, 364
dkinfo, 67, 68, 217
dkinfo command, 68
dkio, 165
dmesg, 217
dname, 217
DNS (Domain Name Service), 139, 140
DNS and NIS+, 140
DNS, definition, 364
document tools, 59
dodisk, 217
Domain Name Service (DNS), 139, 140
domainname, 217
dorfs, 217
dos, 217
dos2unix, 217
driver configuration files, 188
Driver Kernel Interface (DKI), 196
du, 67, 68, 102, 217, 218
dumbplot, 218
dump, 105, 106, 107, 218
 and object files, 159
dump device, default, 87
dumpfs, 218
dumpkeys, 218
dynamic linking, 9
dynamic linking, ld, 154
dynamically configured kernel, 16, 69
dynamically linked applications, 174
dynamically loaded kernel, 187

E

e, 218
echo, 218
ed, 218
edit, 218
edquota, 218
eeprom, 218
EFT (extended fundamental types), 28
egrep, 218
eject, 218
ELF (Executable and Linking Format), 147
ELF, definition, 364
enablenumlock, 218
end user system support cluster, 12
enroll, 218
entire distribution cluster, 12
env, 218
environment-initialization files,
 converting, 34
eqn, 218
error, 219
 /etc, 15, 36, 82, 85, 91, 353
 /etc/acct, 353
 /etc/cron.d, 353
 /etc/default, 92, 354
 /etc/default directory
 files in, 47
 security, 47
 /etc/default/fs, 90
 /etc/dfs, 354
 /etc/dfs/dfstab, 103
 /etc/dfs/fstype, 90
 /etc/exports, 38, 103
 /etc/fs, 354
 /etc/fstab, 15, 35, 37, 38, 82, 91, 92, 93,
 96, 101, 106
 /etc/group, 38, 123
 /etc/inet, 92, 354
 /etc/init.d, 354
 /etc/inittab, 75

/etc/inittab file, 76
 /etc/lib, 354
 /etc/lp, 15, 82, 92, 354
 /etc/mail, 354
 /etc/net, 354
 /etc/netgroup, 38
 /etc/opt, 92, 354
 /etc/passwd, 38, 46, 52
 /etc/printcap, 22, 37, 39, 82, 122
 /etc/rc, 76, 77
 /etc/rc.3, 354
 /etc/rc.boot, 76, 77
 /etc/rc.local, 76, 77
 /etc/rc0.d, 354
 /etc/rc1.d, 354
 /etc/rc2.d, 354
 /etc/rcn.d, 92
 /etc/saf, 92, 354
 /etc/sendmail.cf, 39
 /etc/shadow, 46, 96
 /etc/skel, 53, 354
 /etc/sm, 354
 /etc/sm.bak, 354
 /etc/system, 189
 /etc/tm, 354
 /etc/ttytab, 39
 /etc/uucp, 38, 354
 /etc/vfstab, 93, 94, 101, 106
 /etc/vstab, 91
 etherd, 219
 etherfind, 219
 Ethernet addresses, 141
 EUC, 183
 ex, 219
 excluded features, 9
 Executable and Linking Format (ELF), 147
 executables, building, 156
 expand, 219
 expanded fundamental types, 9
 /export, 354
 /export/home, 354
 exportfs, 103, 219
 exporting file systems, 103
 expr, 219
 extended fundamental types (EFT), 28
 Extended UNIX Code (EUC), 183
 extract_files, 219
 extract_patch, 219
 extract_unbundled, 219

F

false, 219
 fastboot, 219
 fasthalt, 206, 219
 fdformat, 219
 FDFS pseudo file system, 83, 86
 features
 excluded, 9
 for the developer, 8
 for the system administrator, 7
 for the user, 6
 ff, 88
 fgrep, 219
 FIFOFS pseudo file system, 83, 87
 file, 219
 file access, 18
 File Manager, 24
 file system administration, 7
 file system information, saving, 29
 file system types, unsupported, 88
 file systems, 15
 added, 16
 adding, 83
 administration, 99
 backing up, 32, 105
 BFS, 88
 changes to, 15, 82
 commands, 88, 91
 creating, 104
 default, 83, 85
 exporting, 103
 formats, 28

HSFS, 86
 monitoring, 102
 mount point, 90
 NFS, 86
 PCFS, 86
 pseudo, 16
 restoring, 105
 S5, 88
 sharing, 103
 SPECFS, 86
 special device, 90
 syntax of generic commands, 89
 types of, 89
 UFS, 86
 xnamefs, 88

file systems, psuedo
 CACHEFS, 87
 FDFS, 86
 FIFOFS, 87
 LOFS, 86
 NAMEFS, 87
 PROCFS, 86
 SWAPFS, 87
 TFS, 86
 TMPFS, 86

files
 backing up, 27
 changing, 38
 converting, 37
 merging, 36

filio, 165
 find, 102, 219
 finger, 219
 fingerd, 219
 /floppy, 101
 floppys, *See* diskettes
 fmt, 219
 fmt_mail, 219
 fold, 219
 fontedit, 220
 fontflip, 220
 foption, 220
 format, 68, 220

fpa_download, 220
 fparel, 220
 fpaversion, 220
 fpurel, 220
 fpuversion4, 220
 from, 220
 fsck, 88, 104, 105, 220
 fsck-cdrom, 220
 fsdb, 88
 fsirand, 220
 fstyp, 88
 ftp, 220
 ftpd, 220
 fumount, 220
 fusage, 220
 fuser, 220
 fwtmp, 220

G
 gcore, 220
 generic commands, 86
 generic commands, syntax, 89
 generic file system commands, 88
 generic_args, 220
 get, 221
 get_alarm, 221
 get_selection, 221
 getopt, 221
 getoptcvt, 221
 getopts, 221
 gettable, 221
 getty, 221
 gfixtool, 221
GID 14, 20
 gid_allocd, 221
 gigiplot, 221
 glob, 221
 goto, 221
 gpconfig, 221
 gprof, 221

graph, 221
grep, 221
group ID, 20
groups, 221
groups, cylinder, 14
grpck, 221
gxtest, 221

H

halt, 221
halting the system, 74
hashcheck, 222
hashmake, 222
hashstat, 222
head, 222
help, 222
help_open, 222
help_viewer, 222
heterogeneous server, definition, 364
/home, 85
homogeneous server, definition, 364
Host Manager, 20, 119
 information needed to use, 120
 security, 47
hostid, 222
hostname, 222
hostrfs, 222
hp7221plot, 222
hpplot, 222
HSFS file system, 86
htable, 222

I

i386, 222
iAPX286, 222
icheck, 222
iconedit, 222
id, 222
idload, 222
ifconfig, 222
imemtest, 222
implot, 222
in.comsat, 222
in.fingerd, 222
in.ftpd, 222
in.named, 223
in.rexecd, 223
in.rlogind, 223
in.routed, 223
in.rshd, 223
in.rwhod, 223
in.talkd, 223
in.telnetd, 223
in.tftpd, 223
in.tnamed, 223
in.uucpd, 223
indent, 223
indentpro_to_defaults, 223
indxbib, 223
inetboot, 189
inetd, 223
infocmp, 223
init, 75, 76, 223
init program, 132
initialization files, converting, 53
initialization run levels, 77
initialization scripts, location, 92
initialization states, 76
inittab, 75
inline, 223
input_from_defaults, 223
insert_brackets, 224
install, 224
install server, definition, 364
install4x, 112, 115 to 118
 user interface, 116
installboot, 224
installing packages, 12
installing Solaris software, 33
 backup disk partitions, 27

Host Manager, 27
order for networks, 31
overview, 25
preinstallation, 26, 27
what's new, 26

installtxt, 184, 224

instance names, 193

internationalization, 183
character support, 183
commands, 184
libraries, 184
message catalogs, 184

interoperability, 5

interprocess communication, 7

interrupted transmission, 136

intr, 224

introduction, 3

ioctl requests, 165
dkio, 165
filio, 165
mtio, 165
sockio, 165
streamio, 165
STREAMS, 201
supported, 165
termio, 165
termios, 165
transparent, 201
unsupported, 166

iostat, 224

IP address, definition, 364

ipallocald, 224

ipcrm, 224

ipcs, 224

J

job grades, 136
join, 224

K

kadb, 224
kadb(1M), 162

Kerberos security, 49
/kernel, 15, 75, 97, 98, 354

kernel, 73, 83, 98
configuration, 16
contents, 17
debugging, 163
development tools, 164
directories, 17
dynamic configuration, 69
dynamic loading, 187
layout, 17, 188
modules, 83, 98
multithreading, 200

kernel architecture, definition, 365

kernel modules, 15
/kernel/unix, 27, 98

keyenvoy, 224

keylogin, 224

keylogout, 224

keyserv, 224

kgmon, 224

kill, 224

Korn shell, 51, 52

L

labelit, 88, 105, 224

last, 224

lastcomm, 224

lastlogin, 225

ld, 154, 225
link editor, 154

LD_LIBRARY_PATH, 158

LD_RUN_PATH, 158

ldconfig, 225

ldd, 225

leaf drivers, 202

leaf nodes, 202

leave, 225

lex, 225

Library Routine Reference Table, 275 to
346

line, 225
link, 225
link editor, ld, 154
linker, run-time, 158
lint, 147, 225
lint libraries, 147
listen, 225
listen port monitor, 131, 132
ln, 225
load, 225
load_package, 226
loadc, 226
loading server software, 116
loadkeys, 226
local file systems, saving, 29
lockd, 226
lockscreen, 226
LOFS pseudo file system, 86
log files, 134, 137
 accounting, 137
 command, 137
 performance, 137
 security, 137
logger, 226
logical device names, 193
.login, 34
.login, 53
login, 226
login default file, 47
login shell, 57
 Bourne shell, 51
 C shell, 51
 korn shell, 51
logintool, 226
logname, 226
look, 226
lookbib, 226
lorder, 226
lp, 226
LP print service, 121
lpadmin, 37

lpc, 121, 226
lpd, 121, 226
lpq, 121, 226
lpr, 121, 226
lprm, 121, 226
lpstat, 226
lpptest, 226
ls, 102, 226
lsw, 226

M

m4, 227
m68k, 227
mach, 227
machid, 227
Mail, 208
mail, 59, 227
 using, 58
mailrc_to_defaults, 227
mailstats, 227
mailtool, 59, 227
mailx, 59
major version numbers, 158
make, 172, 227
makedbm, 227
MAKEDEV, 76
makedev, 227
Makefiles (make), 172
makekey, 227
man, 228
man command search path, 61
man pages, 60
maps, autofs, 99
MAXUSERS, 189
mc68010, 228
mc68020, 228
mc68881version, 228
mconnect, 228
merging SunOS 4.x and 5.x files, 36
mesg, 228

message catalogs, 184
metadevice information, saving, 30
migrating, advantages, 3
minor version numbers, 158
mkdir, 228
mkfile, 228
mkfs, 88, 104, 228
mknod, 228
mkproto, 228
mkstr, 228
/mnt, 354
moddir variable, 17, 188
modload, 17, 188, 228
modstat, 228
module directory search path, 188
modules, kernel, 98
modunload, 17, 188, 228
monacct, 228
montbl(1M), 184
more, 228
mount, 88, 100, 229
mount point, 90
mount_rfs, 91
mount_tfs, 229
mountall, 88
mountd, 229
mounting, automatic, 99
msgfmt, 184
mt, 229
mtio, 165
multibyte support, 183
multiple OS operation, 112
multiple OS operation, definition, 365
multiple OS software, 112
 convert4x, 112
 discover4x, 112, 113
 install4x, 112
multithreaded (MT) kernel, 8, 200
mutexes, 200
mutual exclusion locks, 200

mv, 229

N

named, 229
NAMEFS pseudo file system, 83, 87
naming devices, 14
naming service
 and the Administration Tool, 20
nawk, 229
ncheck, 88, 229
ndbootd, 229
neqn, 229
netconfig, 229
netmask, definition, 365
netstat, 229
network facilities, 133
Network Information Service (NIS), 139
Network Information Service + (NIS+), 21
 and security, 48
Network Information service + (NIS+)
 and security, 46
Network Information Services +
 (NIS+), 4, 8
newaliases, 229
newfs, 104, 229
newgrp, 229
newkey, 229
nexus drivers, 203
nexus nodes, 203
NFS, 134
NFS file system, 86
NFS security, 46
nfsd, 229
nfsstat, 229
nice, 230
NIS (Network Information Service), 139,
 182
 and SunOS release 5.4, 21
NIS and SunOS 5.x, 33
NIS master server, transition
 information, 33

NIS support, 143
NIS+
 and DNS, 140
NIS+ (Network Information Service
 +), 182
NIS+ (Network Information Service
 +), 21, 141
 benefits, 142
 installing on 4.1 server, 143
 vs NIS, 21, 142
NIS+ (network information service +)
 maps, 133
NIS+, definition, 365
NIS, definition, 365
NISBCP, 143
NIT, 182
nl, 230
nlsadmin, 230
nm, 230
nohup, 230
nroff, 230
nslookup, 230
nsquery, 230
nulladm, 230

O

objdump, 230
objective of this book, xxii
OBP, 188
od, 230
old-analyze, 230
old-ccat, 231
old-clocktool, 231
old-compact, 231
old-eyacc, 231
old-filemerge, 231
old-make, 231
old-perfmon, 231
old-prmail, 231
old-pti, 231
old-setkeys, 231

old-sun3cvt, 231
old-syslog, 231
old-uncompact, 231
old-vc, 231
OLIT, 176
on, 231
ONC, 140
Online DiskSuite, 30
OpenBoot PROM, 188
OpenWindows, 54
/opt, 16, 83, 85, 98, 354
organization of this book, xxii
organizer, 231
overview, 231

P

pac, 231
pack, 231
package
 administration, 12
 commands, 13
 installing, 12
 removing, 12
package information file, pkginfo, 174
package, definition, 365
packaging applications, 174
packaging commands, 175
page, 231
pagesize, 231
partition slice numbers, 66
passwd, 232
password aging, 46, 48
passwords, 46
paste, 232
path, 58
pax, 232
paxcpio, 232
pcat, 232
PCFS file system, 86
pdpl1, 232

perfmeter, 232
 performance log file, 137
 permissions, 58
 pg, 232
 physical device names, 192
 ping, 232
 pkginfo, 113, 174
 plot, 232
 plottoa, 232
 pmadm, 130, 131, 132
 pnp.s386, 232
 pnpboot, 232
 pnpd, 232
 Point-to-Point Protocol (PPP), 134
 port monitors, 130
 portability, 5
 porting applications, 42
 porting device drivers, 199 to 201
 portmap, 232
 PostScript® filters, 59
 PPP (Point-to-Point Protocol), 134
 pr, 232
 praudit, 232
 prc.yupdated, 236
 prctmp, 232
 prdaily, 232
 primary groups, 57
 print command changes, 22
 print management, 21
 print service, LP, 121
 print subsystem, 21
 printenv, 232
 Printer Manager, 20, 22, 121, 122
 printers, 121 to 127

- administration, 125
- and the compatibility package, 124
- commands, 125
- information for Printer Manager, 123
- local and remote, 123
- set up, 122
- setting up, 124

printing, 121
 printing troff, 126
 PrintTool, 22
 /proc, 85
 process management, 7
 PROCFS pseudo file system, 83, 86
 prof, 233
 .profile, 34, 53
 programming tools, 165 to 175
 PROM, booting from, 74
 prototype file, 174
 prs, 233
 prt, 233
 prtacct, 233
 prtconf, 198
 prtvtoc, 68, 69
 prtvtoc command, 68
 ps, 233
 pseudo file systems, 16, 82

- CACHEFS, 82
- FDFS, 83
- FIFOFS, 83
- NAMEFS, 83
- PROCFS, 83
- SWAPFS, 83

pstat, 233
 ptx, 233
 pwck, 233
 pwd, 233
 pwauthd, 233

Q

QuickCheck, 28
 quot, 102, 233
 quota, 233
 quotacheck, 233
 quotaoff, 234
 quotaon, 234

R

ranlib, 234
rarpd, 234
rasfilter8tol, 234
rastrepl, 234
rc, 92, 234
 rc* scripts, 15, 82, 96
 rc.boot, 92
 rc.local, 92
 rc.single, 92
rc.boot, 234
rc.local, 234
rcp, 234
rdate, 234
rdist, 234
rdump, 234
reboot, 234
rebooting the system, 74
recommended reading, xxv
red, 234
refer, 234
references, about, xxii
rehash, 234
related books, xxv
remove_brackets, 234
removing packages, 12
renice, 234
repquota, 234
reset, 234
resource limits, 170
restore, 105, 106, 107, 234
restoring
 from a single tape, 35
 from separate tapes, 35
 system data, 36
 user data, 34
restoring bar files, 105
restoring files, 105
restricted shells, 47
rev, 207, 235
rexcd, 235
rexecd, 235
rfadmin, 235
RFC security, 46
rfpasswd, 235
rfstart, 235
rfstop, 235
rfuadmin, 235
rfudeamon, 235
 .rhosts, 46
ring_alarm, 235
rlogin, 235
rlogind, 235
rm, 235
rm_client, 235
rm_services, 235
rmail, 235
rmdel, 236
rmdir, 236
rmntstat, 236
rmt, 236
roffbib, 236
/, root directory, 353
/ ,root direcotry, 85
route, 236
routed, 236
RPATH, 159
RPC program numbers, 141
rpc.bootparamd, 236
rpc.etherd, 236
rpc.lockd, 236
rpc.mountd, 236
rpc.nisd, 143
rpc.rexd, 236
rpc.rquotad, 236
rpc.rstatd, 236
rpc.rusersd, 236
rpc.rwalld, 236
rpc.showfhd, 236
rpc.sprayd, 236
rpc.statd, 236

rpc.user_agentd, 236
 rpc.yppasswdd, 236
 rpcgen, 236
 rpcinfo, 237
 rrestore, 237
 rsh, 237
 run levels, 76
 run time linker, 158
 runacct, 237
 running SunOS release 4.x
 applications, 43
 rup, 237
 ruptime, 237
 rusage, 237
 rusers, 237
 rwall, 237
 rwho, 237

S

S5 file system, 88
 sa, 237
 SAC (Service Access Control), 23, 130, 132
 sac program, 132
 SAC, definition, 365
 sacadm, 130, 131, 132
 sact, 237
 SAF (Service Access Facility), 22, 23, 130,
 132
 SAF, definition, 365
 savecore, 237
 saving disk configuration information, 28
 saving file system information, 29
 saving system components, 29
 saving system configuration
 information, 32
 /sbin, 15, 76, 82, 85, 96, 355
 /sbin/init, 75
 /sbin/rc, 77
 scalability, 5
 sccs, 237
 SCCS utilities, 173
 sccs-admin, 237
 sccs-cdc, 237
 sccs-comb, 237
 sccs-delta, 237
 sccsdiff, 238
 sccs-get, 237
 sccs-help, 237
 sccs-prs, 237
 sccs-prt, 237
 sccs-rmdel, 237
 sccs-sact, 237
 sccs-sccsdiff, 237
 sccs-unget, 238
 sccs-val, 238
 screenblank, 238
 screendump, 238
 screenload, 238
 script, 238
 scrolldefaults, 238
 SCSI, 197
 SCSI disks, 66
 sdiff, 238
 search path
 directory, modules, 188
 libraries, 157
 link editor, 157
 secondary groups, 57
 security, 8, 45
 Administration Tool, 47
 Database Manager, 47
 features, 46
 Host Manager, 47
 Kerberos, 49
 NFS, 46
 NIS+, 46, 48
 restricted shells, 47
 RFC, 46
 security log file, 137
 sed, 238
 selection_svc, 238
 sendmail, 238

Serial Port Manager, 20
server software, loading, 116
server, definition, 365
Service Access Control (SAC), 23, 130, 132
Service Access Facility (SAF), 22, 23, 130, 132
 overview of, 22
 programs associated with (table), 24
 uses for, 22
serving 4.1.x clients, 112
set_alarm, 238
set4, 238
setkeys, 238
setsid, 238
setup_client, 238
setup_exec, 238
sh, 238
share, 103
shareall, 103
shared libraries, 156, 170
 version numbers, 158
shared libraries, building, 156
sharing file systems, 103
shells and security, 47
shelltool, 238
shift_lines, 239
showfh, 239
showmount, 239
shutacct, 239
shutdown, 78
shutdown, 78, 239
shutdown(, 74
size, 239
skeleton path, 58
skyversion, 239
sleep, 200, 239
slice numbers, 66
slice, disk, 13
snap, 239
sockio, 165
soelim, 239
Solaris software, installing, 25
Solaris, additional features, 6
sort, 239
sortbib, 239
sparc, 239
SPARC XENIX® programs, 9
SPARCServer Manager, 30
SPECFS file system, 86
special device, 90
spell, 239
spellin, 239
spline, 239
split, 240
spool directory, 134
spray, 240
standalone, definition, 366
standards conformance, 4
startup, 240
static linking, ld, 154
streamio, 165
STREAMS, 8, 201
strftime(3C), 184
strings, 240
strip, 240
stty, 240
stty_from_defaults, 240
su, 240
sum, 240
sun, 240
Sun Common SCSI Architecture, SCSA, 197
Sun DDI, 9
sundiag, 240
suninstall, 240
SunOS release 4.1 commands, 205
SunOS release 4.x
 command availability, 41
 executable files in Solaris, 34
 running applications, 43
 support for, 112, 113
 system data, modifying, 38

work environment, 54
SunOS release 4.x clients, 112
SunOS/BSD Source Compatibility
Package, 41 to 42, 65, 125
and mail, 58
SunShield, 50
SunView, 54
applications, 54
sunview, 240
SUNWhinst, 112
sv_acquire, 240
sv_release, 240
SVID, 170
SVR4, 3
swap file, 87
SWAPFS pseudo file system, 83, 87
swapon, 240
swin, 241
switcher, 241
swmtool, 13
symbols, xxv
symorder(1), 241
sync, 241
syntax
generic file system commands, 89
/sys, 15, 82, 98, 99
sysadm, 9
sys-config, 241
sysex, 241
syslogd, 241
system accounting, 7
System Call Reference Table, 249 to 273
system componenets, saving, 29
system configuration file, 189
system configuration information,
saving, 32
system data, modifying, 38
system data, restoring, 36
System File Reference Table, 347 to 351
system information, 7
system parameters, 189

System V file system, 9
System V Interface Definition, 170
system(4), 189
sys-unconfig, 241
syswait, 241

T

t300, 241
t300s, 241
t4013, 241
t450, 241
tabs, 241
tail, 241
talk, 241
tape drives
and device naming, 67
tar, 105, 107, 241
tbl, 241
tcopy, 241
tcov, 241
TCP/IP
administration, 133
and NIS+, 133
tee, 241
tek, 241
tektool, 241
telnet, 242
terminal line settings, 130
/terminfo, 15
terminfo database, 122, 125
termio, 165
termios, 165
test, 242
textedit, 242
textedit_filters, 242
TFS (translucent file system), 82
TFS pseudo file system, 82, 86
tfsd, 242
tftp, 242
tic, 242

time, 242
 time zone, definition, 366
 tip, 242
TI-RPC, 140
 /tmp, 85, 355
TMPFS pseudo file system, 86
 toolplaces, 242
 touch, 242
 tput, 242
 tr, 242
 trace, 163, 242
 traffic, 242
TranScript® filters, 59
translucent file system (TFS), 86
 troff, 22, 242
 trpt, 242
 true, 243
 truss, 163
 tset, 243
 tsort, 243
 tty, 243
 ttymon port monitor, 131, 132
 ttysoftcar, 243
tunable parameters, 189
tunable UUCP parameters,
 overriding, 137
 tunefs, 243
 turnacct, 243
 tvconfig, 243
type changes, xxv
types of file systems, 89
 tzsetup, 243

U

u370, 243
 u3b, 243
 u3b15, 243
 u3b2, 243
 u3b5, 243
UFS file system, 86
 ufsboot, 74, 189
 ufsbootblk, 74
 ufsdump, 106
 ufsrestore, 107
 uid_allocd, 243
 ul, 243
 umask, 243
 umount, 88, 243
 umount_tfs, 243
 umountall, 88
 unadv, 243
 uname, 243
 unblock, 107
unbundled, definition, 366
 uncompress, 243
 unconfigure, 243
 unexpand, 244
 unget, 244
 unifdef, 244
 uniq, 244
 units, 244
 unix, 83
 unix, kernel, 15
 unix2dos, 244
UNIX-to-UNIX Copy, see UUCP
 unlink, 244
 unload, 244
 unloadc, 244
 unpack, 244
 unset4, 244
 unwhiteout, 244
 update, 244
 uptime, 244
User Account Manager, 56
user accounts, adding, 55
user and group management, 8
user data, restoring, 34
user environments, customizing, 53
user ID, 57
user job grades, 136

User Manager, 20
 user name, 57
 users, 244
 using mail, 58
 Using Network Facilities, 133 to 137
 /usr, 15, 82, 85, 97, 355
 /usr file system, 356
 /usr/4lib, 356
 /usr/5bin, 97
 /usr/5include, 97
 /usr/5lib, 97
 /usr/bin, 97, 356
 /usr/bin/mail, 59
 /usr/bin/mailx, 59
 /usr/bin/sunview1, 356
 /usr/ccs, 97, 356
 /usr/ccs/bin, 356
 /usr/ccs/lib, 356
 /usr/demo, 356
 /usr/etc, 97
 /usr/games, 356
 /usr/include, 97, 356
 /usr/kvm, 356
 /usr/lib, 97, 356
 /usr//lib/acct, 356
 /usr/lib/dict, 356
 /usr/lib/iconv, 357
 /usr/lib/font, 357
 /usr/lib/fs, 91, 357
 /usr/lib/libp, 357
 /usr/lib/locale, 357
 /usr/lib/lp, 357
 /usr/lib/mail, 357
 /usr/lib/netsvc, 357
 /usr/lib/nfs, 357
 /usr/lib/pics, 357
 /usr/lib/refer, 357
 /usr/lib/sa, 357
 /usr/lib/saf, 357
 /usr/lib/spell, 357
 /usr/lib/uucp, 357
 /usr/lib/class, 356
 /usr/local, 357
 /usr/net/servers, 357
 /usr/oasys, 357
 /usr/old, 97, 357
 /usr/openwin, 357
 /usr/sadm, 357
 /usr/sadm/bin, 357
 /usr/sadm/install, 357
 /usr/sbin, 97, 357
 /usr/sbin/static, 358
 /usr/share, 358
 /usr/share/lib, 358
 /usr/share/lib/keytables, 358
 /usr/share/lib/mailx, 358
 /usr/share/lib/nterm, 358
 /usr/share/lib/pub, 358
 /usr/share/lib/spell, 358
 /usr/share/lib/tabset, 358
 /usr/share/lib/terminfo, 358
 /usr/share/lib/tmac, 358
 /usr/share/lib/zoneinfo, 358
 /usr/share/src, 358
 /usr/snadm, 97, 358
 /usr/ucb, 358
 /usr/ucb/mail, 59
 /usr/ucbinclude, 358
 /usr/ucblib, 358
 /usr/vmsys, 358
 /usr/xpg2bin, 97
 /usr/xpg2include, 97
 /usr/xpg2lib, 97
 ustar, 244
 uucheck, 244
 uucico, 244
 uucleanup, 244
 UUCP, 134
 .D file, 135, 136
 .P file, 135, 136

/etc/uucp/Config, 135, 137
 /etc/uucp/Grades, 135
 /etc/uucp/Limits, 135, 136
 /var/uucp/.Admin/account, 137
 /var/uucp/.Admin/command, 137
 /var/uucp/.Admin/perflog, 137
 /var/uucp/.Admin/security, 137
 7
 interruption of transmission, 136
 Maxuusched parameter, 136
 Maxuuxqt parameter, 136
 overriding tunable parameters, 137
 uuglist command, 135
 uucp, 244
 uudecode, 245
 uuencode, 245
 uulog, 245
 uuname, 245
 uupick, 245
 uusched, 245
 uusend, 245
 uustat, 245
 uuto, 245
 uux, 245
 uuxqt, 245
V
 vacation, 245
 val, 245
 /var, 15, 82, 85, 97, 355
 /var/adm, 355
 /var/crash, 355
 /var/cron, 355
 /var/lp, 355
 /var/mail, 82, 98, 355
 /var/news, 355
 /var/nis, 355
 /var/opt, 97, 355
 /var/options, 355
 /var/preserve, 355
 /var/sadm, 98, 355
 /var/saf, 98, 355
 /var/spool, 355
 /var/spool/cron, 355
 /var/spool/locks, 355
 /var/spool/lp, 355
 /var/spool/mail, 98
 /var/spool/mqueue, 355
 /var/spool/pkg, 355
 /var/spool/uucp, 355
 /var/spool/uucppublic, 355
 /var/tmp, 356
 /var/uucp, 356
 /var/yp, 356
 vax, 245
 vedit, 245
 version numbering, 158
 vfontinfo, 245
 VFS (virtual file system), 104
 architecture, 68, 86
 file, 93
 interface, 86
 vgrind, 245
 vi, 245
 view, 245
 vipw, 245
 virtual file system (VFS), 104
 architecture, 68, 86
 file, 93
 interface, 86
 vmstat, 245
 /vmunix, 17, 27, 75, 188
 /vol, 16, 83, 85
 volcopy, 89, 105
 vold, 16, 83
 /volfs, 16
 volfs file system, 83
 volume management, 16, 24, 70, 83
 administration, 24
 commands, 72
 File Manager, 24
 vplot, 245

/vstab, 15
vswap, 245
vtroff, 245
vwidth, 245

W

w, 245
wait, 246
wakeup, 200
wall, 246
wc, 246
well-behaved applications, 174
what, 246
whatis, 246
whatis man command, 62
whereis, 246
which, 246
who, 246
whoami, 246
whois, 246
windex man command, 62
write, 246

X

xargs, 246
xget, 246
xgettext, 184
xnamefs file system, 88
xsend, 246
xstr, 246
XView, 176

Y

yacc, 246
yes, 246
yic, 247
ypbatchupd, 246
ypbind, 246
ypcat, 246, 247

ypdump, 247
ypinit, 246
ypmatch, 246
yppasswd, 246, 247
yppush, 247
ypserv, 247
ypset, 247
ypupdated, 247
ypwhich, 247
ypxfr, 247
ypxfrd, 247

