



Solaris 1.x to 2.x Transition Guide

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

Part No: 802-5758-10
August 1997

Copyright 1997 Sun Microsystems, Inc. 2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, SunSoft, SunDocs, SunExpress, and Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was 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.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 1997 Sun Microsystems, Inc. 2550 Garcia Avenue, Mountain View, Californie 94043-1100 Etats-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées du système Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, SunSoft, SunDocs, SunExpress, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REPENDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



Please
Recycle

Contents

Preface xiii

Part I Transition Information for Users and System Administrators

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

Patch Administration 13

Disk Slices (or Partitions) 13

Cylinder Groups 14

Device Naming 15

File Systems 15

Changes to File System Locations and Names 16

Pseudo File Systems 16

Added File Systems 17

Removed File Systems 17

Kernel Configuration 17

Kernel Layout	18
Automounting	18
Mail Administration	20
Admintool	20
Network Information Service Plus (NIS+)	21
Print Subsystem	22
PrintTool	22
Command Changes	22
Service Access Facility	23
Volume Management	24
3. Converting a SunOS 4.x System to the Solaris 2.6 Environment	25
What's New About Installing	25
What to Do Before You Install Solaris Software	26
Saving Disk Partition Information	27
Saving File System Information	27
Saving Metadevice Configuration Information	28
Determining What To Back Up	28
Determining Disk Space Requirements	30
Deciding the Order of Installation for Networks	30
Backing Up Files and File Systems Before You Install	30
Installing Solaris Software	31
Preserve Option	31
Restoring Files and File Systems After You Install	31
Restoring SunOS 4.x File Systems and User Files	31
Restoring SunOS 4.x System Configuration Files	32
4. Using the Compatibility Packages	35
Why Port Applications?	35
SunOS/BSD Source Compatibility Package	36

	Binary Compatibility Package	36
	Using the Binary Compatibility Package to Run SunOS Release 4.x Applications	37
5.	Security	39
	Solaris 2.6 Security Features	39
	/etc/passwd and /etc/shadow Files	40
	/etc/default Files	40
	Restricted Shells	41
	Password Aging Changes	41
	Access Control Lists (ACLs)	42
	Automated Security Enhancement Tool (ASET)	42
	Security Options	43
	Kerberos 4.0 Security	43
	SunSHIELD Package	43
	PAM	43
6.	User Environment Administration	45
	Selecting a Default Shell	45
	Customizing User Environments	47
	Using the SunOS 4.x Work Environment With the Solaris Software	48
	Window Systems	48
	User and Group Administration	49
	User and Group Administration Choices	49
	Adding User Accounts	49
	Using Mail	50
	Using Document Tools	51
	Man Page Organization Differences	51
	Customizing the man Command Search Path	53
	whatis and windex Databases	54

	Using the <code>man</code> Command	54
7.	Device Administration	57
	Device Naming Conventions	57
	Convention for Disks	58
	Convention for Tape Drives	58
	Obtaining Disk Information	59
	<code>df</code> Command	59
	<code>du</code> Command	60
	<code>dckinfo</code> Command	60
	<code>devinfo</code> Command	61
	Adding Devices to the System	61
	Using Volume Management	61
8.	Startup and Shutdown	65
	Booting	65
	<code>boot</code> Command Changes	66
	Booting From the PROM	67
	Summary of Boot Differences	67
	Using the <code>init</code> Command	68
	<code>init</code> Command Changes	68
	Changing System Run Levels	69
	Shutting Down	70
	Changes to the <code>shutdown</code> Command	71
	Using the <code>fasthalt</code> and <code>fastboot</code> Commands	72
	Using the <code>halt</code> and <code>reboot</code> Commands	72
9.	File System Administration	73
	File System Changes	73
	Pseudo File Systems	74
	Added File Systems	75

Default File Systems and Directories	75
Virtual File System Architecture	77
Supported File System Types	77
Unsupported SVR4 File System Types	79
Generic File System Commands	79
Directory and File Changes	82
/dev Directory	82
/etc Directory	83
/sbin Directory	87
/usr Directory	87
/var Directory	88
/kernel Directory	88
/opt Directory	89
/sys Directory	89
Using File System Administration Commands	90
Mounting File Systems and <code>autofs</code>	90
Monitoring File Systems	92
Sharing File Systems	93
Creating New File Systems	94
Checking File Systems	95
Backing Up and Restoring Files	96
10. Setting Up a Solaris 2.6 Server to Support SunOS Release 4.x Diskless/ Dataless Clients	101
Adding SunOS Release 4.x Support to a Solaris 2.6 Server	101
Running <code>discover4x</code>	102
Setting Up the CD-ROM Drive for <code>install4x</code>	103
Running <code>install4x</code>	105
Running <code>convert4x</code>	108

- 11. Managing Printers, Terminals, and Modems 109**
 - Printing 109
 - Summary of Printing Differences 109
 - Print Commands and the Compatibility Package 110
 - Using Printer Commands 110
 - Using SunOS Release 5.6 Printer Administration Commands 111
 - Serial Port Management 113
 - Terminal and Modem Management 113
 - Service Access Facility (SAF) 113
- 12. Network Service Administration 117**
 - Changes to TCP/IP 117
 - Changes to NFS 118
 - PPP 118
 - UUCP 119
 - Checkpoint Restart 120
 - User Job Grades 121
 - Limits File 121
 - Config File 121
 - Log Files 121
- 13. Using Name Services 123**
 - Name Service Switch 123
 - NIS+ 124
 - DNS 124
 - DNS and NIS+ Comparison 124
 - NIS and NIS+ Comparison 125
 - Planning an NIS+ Upgrade 127
- 14. Solaris Common Desktop Environment 129**
 - What Is the Solaris Common Desktop Environment? 129

Developers, End Users, and CDE	130
Overview of the Desktop	130
Front Panel	130
Style Manager	131
File Manager	132
Application Manager	132
Moving From the OpenWindows Environment to CDE	133
Desktop Services	133
Using Windows, Menus, Buttons, and the Mouse in CDE	134
Accessing the Workspace Programs Menu	134
Style Manager and Customizing the Workspace	134
Running OpenWindows Applications in CDE	135
Application Settings and Properties	135
Changing Keyboard Defaults	135
Changing Mouse Defaults	136
Part II Transition Information for Developers	
15. Compilers, Linkers, and Debuggers	139
Compilers	139
Compiler Option Differences	140
Linkers	148
Link Editor Option Differences	148
Building Shared Libraries	151
Building Executables	151
Specifying Library Search Paths	152
Search Path Rules	152
Version Numbering	153
Examples	154
Debuggers	156

	dbx and dbxtool	156
	adb and kadb	156
	kadb Macros	156
	Debugging a Live Kernel	157
	truss Command	157
16.	Tools and Resources	159
	ioctl() Requests	159
	ptrace() Request Values	162
	Libraries	163
	Reorganized Libraries	164
	Shared Libraries	164
	Resource Limits	164
	Using make	167
	Using SCCS	168
	Determining Application Compatibility	168
	Packaging Applications	169
	Packaging Utilities	170
	Toolkits	171
	OLIT	171
	XView	171
	Finding SunOS Release 4.x Tools	171
17.	Networking and Internationalization	177
	Networking	177
	NIS, NIS+	177
	nsswitch.conf File	178
	Network Interface Tap	178
	Sockets	178
	Internationalization	178

	Character Support	179
	Message Catalogs	179
	Locale Database	180
	Commands	180
	Libraries	180
18.	System and Device Configuration	183
	System Configuration	183
	Dynamically Loaded Kernel	183
	Kernel Layout	184
	config Command	184
	/etc/system File	184
	boot Command	185
	Summary of Boot Differences	185
	Reconfiguration Boot	186
	Device Naming From a Developer's Perspective	187
	/devices	187
	/dev	187
	Device Driver Naming	188
19.	Device Drivers and STREAMS	191
	Device Drivers and STREAMS Device Drivers	191
	Device Driver Interfaces	192
	devinfo Command	193
	Porting Considerations	195
	STREAMS	196
	Solaris 2.x Driver Architecture	197
	Device Driver Commands	198
A.	Commands Reference Table	199
	Using the Reference Table	199

	Examples	200
	Commands Reference Table	202
B.	System Calls Reference Table	255
	Using the Reference Table	255
	Examples	256
	System Calls	257
C.	Library Routines Reference Table	287
	Using the Reference Table	287
	Examples	288
	Library Routines	288
D.	System Files Reference Table	381
	Using the Reference Table	381
	System Files	381
E.	/ and /usr File Systems Changes	387
	Layout of the / File System	387
	Layout of the /usr File System	391
F.	Quick Reference for Basic Changes	395
	Summary Tables	395
	Glossary	403
	Index	407

Preface

The Solaris[™] 2.6 operating environment, SunSoft's[™] distributed computing solution, comprises SunOS[™] release 5.6 software with ONC[™], OpenWindows[™], ToolTalk[™], and DeskSet[™] utilities as well as other features. *Solaris 1.x to Solaris 2.x Transition Guide* focuses on the differences between the SunOS release 4.x and SunOS release 5.6 operating systems for people already familiar with the SunOS release 4.x software. This guide also handles other aspects of the Solaris 2.6 operating environment that can help you through the transition.

If you are looking for more information about features now available with the Solaris 2.6 operating environment, see *Solaris User's Guide*.

The system administration tool covered in this book, Admintool, is part of the Solaris 2.6 product and can be used only for local system administration. System administration tools used to manage a network of systems are provided with the Solstice family of products.

Note - The term "x86" refers to the Intel 8086 family of microprocessor chips, including the Pentium and Pentium Pro processors and compatible microprocessor chips made by AMD and Cyrix. In this document the term "x86" refers to the overall platform architecture, whereas "*Intel Platform Edition*" appears in the product name.

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 2.6 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 5.6 operating environments to make your transition to the Solaris 2.6 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 2.6 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 2.6 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, discusses the benefits of migrating to the Solaris operating environment and summarizes the main differences between SVR4 and the Solaris operating environment.
- Chapter 2, is an overview of some of the principal changes between the SunOS release 4.x and SunOS release 5.6 environments. It provides background for topics in subsequent chapters, focusing on procedures, tools, and concepts that have changed between releases.
- Chapter 3, 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 2.6 operating environment.

- Chapter 4, 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 2.6 operating environment.
- Chapter 5, describes the major differences between SunOS release 4.x and Solaris 2.6 security, and points out how those changes might affect system administration procedures.
- Chapter 6, 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, explains SunOS release 5.6 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, describes changes to procedures for booting and shutting down a system.
- Chapter 9, 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, discusses setting up servers for clients. It describes three programs—`discover4x`, `install4x`, and `convert4x`—that work together to help prepare a Solaris 2.6 server to serve SunOS release 4.x clients.
- Chapter 11, describes how to set up and administer printers after you install Solaris 2.6 software and changes to printer commands. It also describes terminal and modem management using `Admintool` and the Service Access Facility (SAP).
- Chapter 12, outlines changes to the network facilities, TCP/IP and UUCP.
- Chapter 13, discusses NIS+ and the domain name system (DNS), and compares NIS+ to NIS and DNS.
- Chapter 14, describes the Common Desktop Environment (CDE) and how to make the transition from the OpenWindows environment to CDE.

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, discusses which capabilities have been added to or removed from compilers, linkers, and debuggers.
- Chapter 16, 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 2.6 packaging capabilities, and how to find SunOS release 4.x tools.
- Chapter 17, discusses Solaris 2.6 networking features as they relate to the programming environment. It also describes improved internationalization features.
- Chapter 18, 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, discusses device drivers issues such as changes to device driver interfaces, the `devinfo` command, porting considerations, `STREAMS`, and the Solaris 2.6 driver architecture.

Reference Appendixes

The following appendixes comprises 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, compares SunOS release 4.x and SunOS release 5.6 commands.
- Appendix B, compares SunOS release 4.x and SunOS release 5.6 system calls.
- Appendix C, compares SunOS release 4.x and SunOS release 5.6 library routines.
- Appendix D, compares SunOS release 4.x and SunOS release 5.6 system files.
- Appendix E, compares SunOS release 4.x and SunOS release 5.6 system files.
- Appendix F, is a quick reference for changes in common commands, files and directories, and daemons and standard processes.

Ordering Sun Documents

The SunDocs™ program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals using this program.

For a list of documents and how to order them, see the catalog section of the SunExpress™ Internet site at <http://www.sun.com/sunexpress>.

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. <code>system% You have mail.</code>
AaBbCc123	What you type, contrasted with on-screen computer output	<code>system% su</code> <code>Password:</code>
<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:

TABLE P-1 Typographic Conventions (continued)

Typeface or Symbol	Meaning	Example
%	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* section.

Related Books

For more information the Solaris 2.6 operating environment, see the following documentation:

- *Solaris User's Guide*
- *Solaris Advanced User's Guide*
- *Installation Instructions for Solaris 2.6 (SPARC Platform Edition)*
- *System Administration Guide*
- *NIS+ Transition Guide*
- *NFS Administration Guide*
- *Solaris Naming Administration Guide*
- *Solaris Naming Setup and Configuration Guide*
- *TCP/IP and Data Communications Administration Guide*
- *SunOS Reference Manual*

- *Binary Compatibility Guide*
- *Source Compatibility Guide*
- *Developer's Guide to Internationalization*
- *Multithreaded Programming Guide*
- *Linker and Libraries Guide*
- *Programming Utilities Guide*

Getting Help from Sun Microsystems' WWW Site

You can get additional Solaris transition information by accessing the following URL:

`http://www.sun.com/smcc/solaris-migration/index.html`

The Solaris Migration Initiative home page is a central point for the distribution of tools, documentation, and information to aid you in your migration to Solaris 2.x. You can always find the most current resources and pointers here.

PART I Transition Information for Users and System Administrators

You can use this part of the guide to help install Solaris 2.6 software, and to understand changes to the local computing environment and to routine tasks.

Introduction

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

- “Advantages of Migrating to the Solaris Operating Environment” on page 3
- “Comparison of SVR4 and the Solaris Operating Environment” on page 6

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 5.6, a full 32-bit operating system based on UNIX System V Release 4 (SVR4)
- Cross-functional compatibility, which enables the SunOS release 5.6 software to run on SPARC[™] as well as 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
- Common Desktop Environment, a desktop graphical interface. This window environment helps you organize and manage your work. The desktop provides windows, workspaces, controls, menus, and a Front Panel allowing simple access to Mail, File Manager, Printers, Image Tool, Calendar Manager, and others
- Calendar Manager, a time management application that displays appointments and ToDo items at a glance and offers a multibrowse feature that makes scheduling among a group easy
- File Manager, a graphical and intuitive way to navigate to local and remote file systems
- Image Tool, which enables you to load, view and save images of over 40 different formats
- Audio, a new, Motif-based audio application for playing and recording AU, WAV, and AIFF files.
- OPEN LOOK Admintool, the base for local system administration
- Installation GUI for easing install and update
- Log-based file systems on servers
- Advanced architecture that includes fully symmetric multiprocessing and sophisticated multithreading
- Real-time priority scheduling and a fully pre-emptible kernel, providing the benefits of open systems while meeting the requirements of control applications
- Network Information Services Plus (NIS+), 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, for sending messages that incorporate audio, graphics, and embedded files
- Java Virtual Machine[™], provides access to the Java platform for the Solaris operating environment

- WebNFS, makes it possible to make a file system accessible through a Web browser.
- AnswerBook2™ Viewer, Sun's premier online documentation system that uses a web-browser-based interface

Portability, Scalability, Interoperability, and Compatibility

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

Portability

The SunSoft SunOS product 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, enabling 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:

- *Workspace manager.* Provides basic window management services (open, close, move, and so on), as well as tools that enable user to customize their workspace.
- *Desktop integration services.* These include ToolTalk, drag and drop, and cut and paste, providing the foundation that enables 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.* Enables 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 help you control system scheduling. Using these utilities, administrators can generate

reports on performance, logins, disk access locations, and seek distances to better tune system performance. In addition, you can change the system run level, kill active processes, time the execution of commands, and change the default scheduling priorities of kernel, timesharing, and real-time processes.

- *System accounting.* The accounting utilities enable system administrators to track system usage by CPU, user, and process for better resource allocation.
- *System information.* These utilities report system memory and system configuration. The system administrator can use the utilities to change the names of the systems and the network node.
- *User and group management.* With these utilities, a system administrator can create and delete entries in group and password databases, specify default home directories and environments, maintain user and system logins, and assign group and user IDs. The utilities support both primary and supplementary user groups.
- *Admintool.* Admintool, which runs under the OpenWindows environment, provides system management facilities to help add hosts, manage the network, and perform many other routine tasks on local systems.
- *Auto configuration.* The Solaris operating environment has a dynamic kernel, which means that it loads drivers and other modules into memory when the devices are accessed. You no longer need to rebuild the kernel after installation, nor must you add or remove drivers.
- *Network Information Services Plus (NIS+).* An upward-compatible version of the NIS name service with simpler hierarchical administration, improved security, and faster updates.
- *Installation.* The Solaris operating environment has an install GUI to ease installation or upgrades. Automatic installations and upgrades are also possible over the network.
- *Security.* The automated security enhancement tool (ASET) is a utility that improves security by allowing system administrators to check system file settings including permissions, ownership, and file contents. ASET warns users about potential security problems and, where appropriate, sets the system file permissions autonomically according 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 flexible framework for character input and output (I/O) that has been implemented throughout SVR4. It is easily customized for applications.
- *Expanded fundamental types*. ID data types (`uid`, `pid`, device IDs, and the like) 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 their limitations compared to the UNIX file system. The SVR4 boot file system was not included because of its maintenance burden when compared to the SunOS traditional boot model.

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 that simplify the administration of distributed systems across a network. The Solaris operating environment provides the utilities and configuration directories that underlie the SVR4 *sysadm* utility but not the *sysadm* utility itself.

Overview of Major Changes

As you use the Solaris 2.6 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 information for topics in subsequent chapters. Some topics receive sufficient coverage here, while others require more in-depth technical background. In the latter case, the text directs you to a chapter that more fully describes the changes.

- “Software Packages and Clusters” on page 12
- “Disk Slices (or Partitions)” on page 13
- “Device Naming” on page 15
- “File Systems” on page 15
- “Kernel Configuration” on page 17
- “Automounting” on page 18
- “Admintool” on page 20
- “Network Information Service Plus (NIS+)” on page 21
- “Print Subsystem” on page 22
- “Service Access Facility” on page 23
- “Volume Management” on page 24

Software Packages and Clusters

Solaris 2.6 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 plus additional software.

- *Core System Support* is the minimum software configuration; it contains only the software necessary to boot and run the Solaris 2.6 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 2.6 operating environment. Compilers and debuggers are not included in the Solaris 2.6 operating environment.
- *Entire Distribution* contains the entire Solaris 2.6 environment.

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

Package Administration

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

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

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

Graphical User Interface (`admintool`)

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

Use `Admintool` to:

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

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

Command-Line Utilities

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

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

Patch Administration

The `patchadd(1M)` and `patchrm(1M)` commands are used to install and remove patches from a Solaris 2.x system. You can add one or more patches to a system, client, service or net install image.

See `patchadd(1M)` and `patchrm(1M)` for more information.

Disk Slices (or Partitions)

A single range of contiguous blocks or a physical subset of a disk is known as a disk *partition* in the SunOS release 4.x software. In the SunOS release 5.x software, a physical subset of a disk is known as a disk *slice*. Before you can create a file system on a disk, you must format and divide it into slices. This is usually done when the Solaris release is installed using the Solaris 2.x installation program. See *System Administration Guide* if you need to install and format a disk after installation.

Note - In some Solaris documentation, Solaris slices are still referred to as “partitions”. The Solaris 2.x documentation distinguishes between `fdisk` partitions (for Intel systems) and the divisions within an `fdisk` partition, referred to interchangeably as slices or partitions.

See *System Administration Guide* for information about Solaris `fdisk` partitions.

A slice can be used as a raw device for swap space or to hold one and only one UFS file system, unless you are using a product like Solstice DiskSuite . Table 2-1 describes how disk slices can be set up on each Solaris 2.x platform.

TABLE 2-1 Slice Differences on Platforms

SPARC	Intel-based
The whole disk is devoted to the Solaris operating environment.	The disk is divided into four <code>fdisk</code> partitions, one per operating environment.
The disk is divided into eight slices, numbered 0-7.	The Solaris <code>fdisk</code> partition is divided into 10 slices, numbered 0-9. Only 0-7 can be used to store user data.

See *System Administration Guide* for a description of customary disk slice assignments for each platform.

Cylinder Groups

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

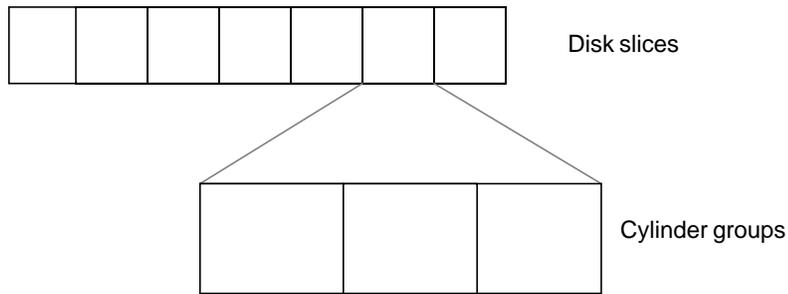


Figure 2-1 Disk Slices and Cylinder Groups

Device Naming

SunOS release 5.6 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 5.6 conventions are slightly different from AT&T SVR4 device names because SunOS release 5.6 allows only eight user-configurable slices on a disk.

In addition, special device files are now stored in the hierarchical `/devices` directory, with symbolic links to the hierarchical `/dev` directory, which is used by administrators and users to access devices. The `/dev` directory contains subdirectories, such as `/dev/dsk/*`, used to access disk devices, and `/dev/rdsk/*`, used to access raw disk devices. For more information, see “Device Naming Conventions” on page 57. For discussions on device naming conventions, see “Device Naming From a Developer’s Perspective” on page 187.

File Systems

SunOS release 5.6 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.

“File System Changes” on page 73, describes file system changes. *System Administration Guide* describes file system concepts and administration in detail.

Changes to File System Locations and Names

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.
- The `/etc/vfstab` file replaces `/etc/fstab`.
- The `/etc/lp` directory replaces `/etc/printcap`.
- The SunOS release 5.6 `/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 5.6 `/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.
- 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 SunOS release 5.6 core kernel is called `genunix`, and the kernel modules are stored in the `/kernel`, `/usr/kernel`, `/platform`, and `/usr/platform` directories.

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 the SunOS release 5.6 software.

The SunOS release 5.6 pseudo file systems are:

- `CACHEFS` pseudo file system – can be used to improve performance of slow devices such as a CD-ROM drive.
- `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)` man 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 – used mostly by STREAMS for dynamic mounts of file descriptors on top of files.
- SWAPFS pseudo file system – 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 5.6 directory structure:

- The optional `/opt` file system, which 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, which provides the default file system for the Volume Management daemon, `vold(1M)`. See the `volfs(7)` man page.

Removed File Systems

Support for the RFS file system type has been removed.

Kernel Configuration

Unlike in the SunOS release 4.x software, the SunOS release 5.6 kernel is dynamically configured. This means that you no longer need to rebuild it manually when you make changes to the system configuration.

Starting with release 5.5 of the SunOS software, the kernel and its modules were separated into platform-independent and platform-dependent objects. The platform-independent kernel consists of a small static core, called `/kernel/genunix`, and its dynamically loadable kernel modules are stored in the `/kernel` and `/usr/kernel` directories if they are platform independent, and `/platform` and `/usr/platform` if they are platform independent. See *System Administration Guide* for a description of the platform-dependent directories and their contents.

Drivers, file systems, STREAMS module, 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. The `modunload(1M)` command 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, 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 platform-independent and platform-dependent directory hierarchy. By default, the directory hierarchy is:

- `/kernel`
- `/usr/kernel`
- `/platform`
- `/usr/platform`

The directory search path for modules can be set by the `moddir` variable in the `/etc/system` file. Typically, `/kernel/genunix` is the first portion of the kernel to be loaded. See `system(4)` and `kernel(1M)` for more information.

Automounting

A new version of the automounter, called AutoFS, has been included. In the SunOS 4.X releases, the automounter mounted everything under `/tmp_mnt` and used symbolic links to redirect the lookups. AutoFS allows for file systems to be mounted in place (for instance, `/home`).

In SunOS 4.X, the maps for the automounter were named `auto.master` and `auto.home`. For Solaris 2.6, these maps have been renamed to `auto_master`, `auto_home`, and so on. The NIS+ name service, which is included with the release, requires this change. A default copy of these maps is included in the release, so that the AutoFS service is started when the system is booted. The SunOS 4.X releases did not include the maps, so additional installation steps were required.

The Solaris 2.6 release provides the ability to select the name service that is being used through `/etc/nsswitch.conf`. The automount entry can be changed to select local files, NIS+, NIS or some combination of these.

Earlier releases supported a home directory naming convention like: `/home/server/login`. With the AutoFS maps it is much easier to use `/home/login` for each entry. This new naming convention also provides for location independence. The old convention can still be used, but once a transition to using the AutoFS maps has been made, it will be easier to administer the shorter paths.

The following paths were reserved for use by AutoFS:

- `/net` - for mounting file systems from a known host
- `/home` - for mounting the home directory of a known user
- `/xfn` - for mounting file systems which support the X/Open XFN standard

On home directory servers, the actual home directories should be moved to `/export/home` rather than `/home`, so that they do not conflict with the automounter directory structure. This also means that you cannot mount file systems on `/home` while the automounter is running.

The AutoFS software now has two programs. The first program is `automount` that runs at boot time to establish AutoFS mount points. This command can also be run anytime by superuser to change the mount points. The second command is `automountd` which is a stateless daemon which answers AutoFS file system mount and unmount requests. These two programs replace the 4.1.X `automount` daemon.

The automount daemon is now fully multi-threaded. Multiple automatic mount requests can be serviced concurrently, which makes AutoFS more reliable. In short, one mount request could block connecting to a slow server, while a second request is processed without waiting.

The Solaris 2.6 release supports browsability of indirect AutoFS maps. All mountable entries under an AutoFS mount point (for example, `/home`) are now visible without the overhead of mounting them first.

Also provided is improved on-demand automounting of hierarchically related file systems. Previous releases would automount an entire set of file systems if they were hierarchically related (for example, `/net/server`) even if only one of the file systems was referenced. The file system that is referenced will be dynamically mounted without mounting all of the other file systems in the hierarchy. Other file systems will be mounted when they are individually referenced.

See “Mounting File Systems and `autofs`” on page 90 for more detailed information. Also, *NFS Administration Guide* describes how to use AutoFS.

Mail Administration

The version of `sendmail` that is included on the release is Version 8 compatible. The new version fixes some security holes and includes several improvements to Version 5. Several extensions to the standard BSD release have been added, including name service switch and NIS+ support.

To further support NIS+, a new command, `aliasadm`, has been included. The command will aid in the administration of NIS+ alias lists.

The mailbox spooling directory has been moved from `/var/spool/mail` to `/var/mail`. A new directory, `/var/mail/:saved`, is used for creating locks and temporary files by the `mailx` program. Also, the mail configuration files are now all located in `/etc/mail`. The new directory includes the `aliases` and the `sendmail.cf` files.

The mailbox locking mechanism has been enhanced so that Solaris 2.6 clients can safely mount mailboxes from both Solaris 2.X and SunOS 4.X mail servers. This enhancement eases administration of mail, especially in large sites.

In the Solaris 2.6 release, `/usr/bin/mailx` supersedes `/usr/ucb/mail`. The `mailx` program has been enhanced to behave the same way as the SunOS 4.x version of `/usr/ucb/mail`. The `/usr/ucb/mail` file is now a symbolic link to `/usr/bin/mailx`.

In SunOS 4.X releases, a program called `sendmail.mx` was used in DNS sites to access mail exchange records. The new version of `sendmail` includes the needed functionality and can be configured through `/etc/nsswitch.conf`.

Mail Administration Guide describes the administration of `sendmail`.

Admintool

One of the major changes between SunOS release 4.x and SunOS release 5.6 that affects system administration is the availability of Admintool to perform basic system administration tasks. This tool employs a graphical user interface to simplify tasks, such as managing users, hosts, printers, and serial devices, on local desktop systems.

Admintool applications enable you to manage the following tasks on a local system:

- System database files such as `aliases` and `netmasks`

- User account and group information, including tasks such as adding users and groups, modifying password aging features, and removing user account information
- Printer setup for local and remote printers
- Terminal and modem setup
- Package management

Using a graphical user interface (GUI) like Admintool to perform 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
- The application programs interact with appropriate system daemons and notify you when the two are out of sync

Note - You do not need to be root to start Admintool but you do need to be a member in the `sysadmin` group (GID=14). Use the `groups(1)` command to display your group membership.

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

```
$ admintool &
```

Network Information Service Plus (NIS+)

NIS+ is the preferred network information service for Solaris networks. Solaris networks can also use NIS either as an alternative to NIS+ or as a supplement to NIS+.

NIS+ is a name service built on top of the ONC transport-independent remote procedure call (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, `/etc`, or DNS.
- You can use AdminSuite to make changes to NIS+ tables for adding, modifying, deleting, and searching for information.
- NIS+ enables you to create and maintain an enterprise-wide name service, even across geographically separated sites connected by WAN links.
- You can use the NIS+ backup and restore feature to quickly and easily preserve your name space data set. This feature can also be used to quickly bring additional replica servers on line.

See Chapter 13, in this guide and *NIS+ Transition Guide* and *NFS Administration Guide* for more information.

Print Subsystem

The print management commands have changed between the SunOS release 4.x and Solaris 2.6 operating environments. In the Solaris 2.6 operating environment, you can use command-line procedures or Admintool to set up printers, and you can use administrative commands or the PrintTool to control print jobs.

See Chapter 11, and *System Administration Guide* for more information.

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

PrintTool

PrintTool is a software tool available in the Solaris 2.6 user environment. It provides a graphical user interface within OpenWindows or CDE 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`
- `lpstat(1)` replaces `lpq`
- `cancel(1)` replaces `lprm`

- `troff(1)` requires a printer name
- `TEX`, `pstat` (C/A/T), and raster image filters are not available in the Solaris 2.6 environment

The `lp` service consists of several *daemons*, or processes, that monitor system work, a hierarchy of configuration files in the `/etc/lp` directory, and 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 so you can easily 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 2-2 illustrates the SAF control hierarchy. The `sacadm` command is used to administer the SAC, which controls the `ttymon` and `listen` port monitors.

TABLE 2-2 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>	Monitors serial port login requests
	<code>listen</code>	Monitors requests for network services
Port monitor service administrator	<code>pmadm</code>	Controls port monitor services
Services	<code>logins</code> , <code>remote procedure calls</code> , and so on	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 11 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 CDs and diskettes.

CDE OpenWindows File Manager has been modified to use Volume Management to provide immediate user access to CDs and diskettes with file systems on them. 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.

For more information, see "Using Volume Management" on page 61 in Chapter 7.

Converting a SunOS 4.x System to the Solaris 2.6 Environment

Converting a SunOS 4.x system to the Solaris 2.6 environment is a three-phase process that includes pre-installation (backing up data), installing the Solaris environment, and post-installation (restoring data).

This chapter provides information about the pre-installation and post-installation phases for a single system or an entire network. See Chapter 10, for information about creating an environment that serves both Solaris 2.6 and SunOS release 4.x clients.

- “What’s New About Installing” on page 25
- “What to Do Before You Install Solaris Software” on page 26
- “Backing Up Files and File Systems Before You Install” on page 30
- “Installing Solaris Software” on page 31
- “Restoring Files and File Systems After You Install ” on page 31

What’s New About Installing

The Solaris 2.6 environment introduces a number of changes in the way you install software on systems, which makes it different than installing the SunOS 4.x software. These include:

- The Solaris 2.6 software is distributed on compact disc (CD) only. This means you must have access to a CD-ROM drive before you can install the software. For systems without local CD-ROM drives, you can set up a system that has a CD-ROM drive to act as an install server on the

network. For more information about network installations, see *Solaris Advanced Installation Guide*.

- The Solaris 2.6 software is bundled into modules called *packages*. You can select which packages to install on your system and control the amount of space each installation requires.

Also, related packages are grouped into *clusters*. This means that you can select a cluster to install without having to select each package separately.

- Solaris 2.6 installation also provides a set of software groups, which are groups of packages and clusters for typical users (for example, there is an end-user software group). 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 the Solaris 2.6 software in a limited environment for testing. You can add or remove packages later as you gain more experience with the system.
- The Solaris 2.6 environment includes architecture-specific kernels rather than the generic kernel configuration provided in earlier SunOS software releases. You will find the installed kernel in `/kernel` instead of `/vmunix`.
- The Solaris 2.6 installation program guides you step-by-step through the installation process.
- The Solaris 2.6 environment provides custom JumpStart™ technology to automate installations. This can save you time when you need to install many systems. For more information, see *Solaris Advanced Installation Guide*.

What to Do Before You Install Solaris Software

Converting a SunOS 4.x system to the Solaris 2.6 software involves more than just running the Solaris installation program and loading the software. Usually, there is data on the SunOS 4.x system that needs to be transferred to a Solaris 2.6 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. Because the device naming conventions are different in the Solaris 2.6 operating environment, you might inadvertently choose the wrong disk when you install the Solaris 2.6 software. Backing up the file systems before you begin the installation procedure offers some protection should this occur.

For information about device naming conventions, see “Device Naming Conventions” on page 57.

Note about file system formats:

- If the Solaris 2.6 Extended Fundamental Types (EFT) are not used, the SunOS release 4.x file system format is upwardly compatible with, and in some cases identical to that used in, the software.
- If you are running SunOS 4.1.1 software with QuickCheck or Backup Copilot[™] utilities installed or the SunOS 4.1.2 software, the file system formats are identical.
- If you are running SunOS 4.1.1 software without QuickCheck or Backup Copilot utilities, SunOS 4.0.x, or SunOS 4.1 software, 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 of the system's existing disk partitions. It can serve as a reference for many decisions that are made about configuring the Solaris 2.6 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.

2. Save the disk partition information.

To obtain the partition information encoded on each disk, use the `dkinfo(8)` command. You can pipe the output to a printer or to a file that you can save to another system.

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.”

Saving 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`. Before proceeding, you should make a printed copy of the `/etc/fstab` file to help you construct the Solaris 2.6 file.

Saving Metadevice Configuration Information

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

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

If the system is running SPARCserver Manager or Solstice DiskSuite utilities, you should save the metadevice configuration information before installing Solaris 2.6 software. This enables you to recover the state of the metadevices when you install Solaris 2.6 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 the Solstice DiskSuite product.

Determining What To Back Up

You should create a list of the SunOS 4.x files and file systems that you want to back up and restore after installing Solaris 2.6 software.

Making a List of System Components to Back Up

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)

Making a List of Files and File Systems to Back Up

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.
- 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.

Making a List of SunOS System Configuration Files to Back Up

There are a number of SunOS 4.x system configuration files that can be merged or converted for the Solaris platform. Use the example list that follows to help select the system configuration files you want to back up.

Note - The list contains suggestions. You should study the items carefully and add to or delete from the list 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`

Determining Disk Space Requirements

Make a list of how much disk space each file system that you want to move to the Solaris 2.6 upgrade, uses. Refer to this list when installing the Solaris 2.6 software, since you can partition disk space for your SunOS 4.x file systems when running the Solaris 2.6 installation program.

Deciding the Order of Installation for Networks

If you are converting a network of SunOS 4.x systems to the Solaris 2.6 software, decide the order of the systems to convert to maximize convenience for the users. For example, you might want to convert all client systems before you convert any servers. The first system you convert 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 4.x and Solaris 2.6 systems, and part of your planning should involve determining priorities. For example, you may want to convert one domain and use it for system administration testing and for porting internally developed applications before you convert the entire network environment.

Backing Up Files and File Systems Before You Install

Once you decide which files or file systems you need to back up from the SunOS 4.x system, you can use the standard commands and procedures given in the SunOS 4.x documentation to do backups. The command you use depends on whether the tape drive is local or remote. No matter how you plan to handle the data transfer, it is still a good idea to back up all disk partitions by doing full dumps before you begin the installation process.

Installing Solaris Software

Install the Solaris 2.6 software on the server or standalone system using the software installation procedures given in *Installation Instructions for Solaris 2.6 (SPARC Platform Edition)* or *Installation Instructions for Solaris 2.6 (Intel Platform Edition)*.

Preserve Option

The Solaris 2.6 installation program has a preserve screen that enables you to preserve existing file systems during installation. This is a good way to preserve any SunOS 4.x file systems so you don't have to restore them.

If you cannot preserve a SunOS 4.x file system or you choose not to (because you want to change how the system's disks are partitioned), you should create new file systems with sufficient disk space for the SunOS 4.x file system that you want to restore (using the disk space requirements you recorded earlier). Then you can restore the SunOS 4.x file systems into the new file systems after Solaris is installed.

Restoring Files and File Systems After You Install

This section describes issues related to restoring SunOS 4.x files and file systems you backed up before installing the Solaris 2.6 software.

Restoring SunOS 4.x File Systems and User Files

You can restore the SunOS 4.x file systems that you could not or chose not to preserve into the new file systems you created during the Solaris 2.6 installation. For information about backup and restore procedures, see *System Administration Guide*.

Note - Before proceeding make sure that the target slice is large enough to accommodate the file system being restored.

Restore any SunOS 4.x user files that you backed up, and copy them to the new system.

Restoring SunOS 4.x System Configuration Files

First, you must restore the SunOS 4.x system configuration files to a temporary directory on the Solaris 2.6 system. After the information is back on the system in the temporary directory, you need to make it available in the Solaris 2.6 operating environment. Some of the data can just be merged into the files, while some types of data must be converted to new formats.

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 2.6 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 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.*` files 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 2.6 operating environment. See Appendix D, to examine SunOS release 4.x files for changes.

- 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 2.6 operating environment. Information from these files must be extracted and manually converted in the Solaris 2.6 environment. See Appendix D, to examine SunOS release 4.x files for changes.



Caution - 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 2.6 software.

You must change the following files before merging the data onto the Solaris 2.6 system:

- `/etc/uucp` - There have been some changes to the UUCP system. The `Config`, `Grades`, and `Limits` files are new in the Solaris 2.6 operating environment. The files `Devconfig`, `Devices`, `Dialcodes`, `Dialers`,

Permissions, Poll, Sysfiles, and systems are the same in the Solaris 2.6 operating environment as they were in the SunOS release 4.x software. These files can be merged together. There are also several SunOS release 4.x files that are not used in the Solaris 2.6 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 2.6 operating environment.
- `/etc/netgroup` – There is no `/etc/netgroup` file in the SunOS release 5.6 environment.
- `/etc/exports` – File systems to be shared on the network under the Solaris 2.6 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 2.6 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 the SunOS release 4.x software. However, user passwords are now stored in the `/etc/shadow` file. Refer to the `passwd(4)` and `shadow(4)` man pages for additional information.
- `/etc/sendmail.cf` – The format of `sendmail.cf` is the same as that under the SunOS release 4.x structure. The location of the file is now `/etc/mail/sendmail.cf`.
- `/etc/ttytab` – Under the SunOS release 4.x system, `ttytab` was used to control serial ports and the characteristics of the terminals on those serial lines. Under the Solaris 2.6 operating environment, the Service Access Facility is used to configure this capability.
- `/etc/printcap` – Under the Solaris 2.6 operating environment, printers are configured using the SunOS release 5.6 LP print service. See *System Administration Guide* for additional information.

Using the Compatibility Packages

The SunOS release 5.6 software is neither source nor binary compatible with the SunOS release 4.x software. This means that SunOS release 4.x programs and user applications based on those releases may not run correctly under the Solaris 2.6 operating environment. Compatibility packages make it possible for these programs to run on a Solaris 2.6 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 2.6 operating environment.

- “Why Port Applications?” on page 35
- “SunOS/BSD Source Compatibility Package ” on page 36
- “Binary Compatibility Package ” on page 36

Some SunOS release 4.x commands are not available in the Solaris 2.6 operating environment. Others exist, but have changed. For information about changes to SunOS release 4.x commands in the Solaris 2.6 operating environment, see Appendix A.

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 2.6 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 2.6 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 2.6 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 5.6 interfaces. These interfaces provide a familiar SunOS environment while your environment and applications are migrating to the SunOS release 5.6 software. To use these interfaces, you must either specify the full path name or modify your `PATH` environment variable. When modifying your `PATH` environment variable, note that `/usr/ucb` should precede `/usr/bin`.

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

Binary Compatibility Package

The Binary Compatibility Package is an optional package available with the Solaris 2.6 operating environment. The package allows existing SunOS release 4.x applications, both statically and dynamically linked, to run under the Solaris 2.6 operating environment without modification or recompilation. It handles most binary interface discrepancies between the two releases transparently. This results in a Solaris 2.6 operating environment where SunOS release 4.x applications can run properly.

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

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

The Binary Compatibility Package allows most applications to run under the Solaris 2.6 operating environment, making them available for use before they are ported to SunOS release 5.6. With this package, well-behaved application binaries based on SunOS release 4.x system software will run under the SunOS release 5.6 software without modifications or recompilation.

The Binary Compatibility Package is intended for end-user environments, not for use as a development environment. All SunOS release 5.6 application development should be done under the base SunOS release 5.6 environment.

Security

Security for the Solaris 2.6 operating environment combines several features from SunOS release 4.x and AT&T SVR4 with capabilities added specifically for the new environment. 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 2.6 operating environment security, and points out how those changes may affect system administration procedures. *System Administration Guide* describes the administration and use of these features more fully.

- “Solaris 2.6 Security Features” on page 39
- “Password Aging Changes” on page 41
- “Automated Security Enhancement Tool (ASET)” on page 42
- “Security Options” on page 43

Solaris 2.6 Security Features

Most of the security features from SunOS release 4.x systems are also available in the Solaris 2.6 operating environment. These include:

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

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

Security for local SunOS release 5.6 systems includes storing encrypted passwords in a separate file, controlling login defaults, and restricted shells. Equivalent NIS+ security, described in *NIS+ Transition Guide* and *NFS Administration Guide*, controls network-wide access to systems.

The subsections below summarize security features under local system control.

`/etc/passwd` and `/etc/shadow` Files

The SunOS release 5.6 `passwd` command stores encrypted versions of passwords in a separate file, `/etc/shadow`, and allows only root access to it. This prevents general access to the encrypted passwords that formerly appeared in the `/etc/passwd` file, which anyone could read.

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

`/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 the 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 for a description of these shells.

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

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

Password Aging Changes

The SunOS release 5.6 system features password aging. This feature assigns a limited lifetime to each user password to maintain password secrecy. As a password reaches the end of its life, the password owner is notified and prompted to select a new one.

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

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

A system administrator can also set up password aging.

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

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

For more information on `passwd` and `nispasswd`, see the command tables in Appendix D.

Access Control Lists (ACLs)

Access control lists (ACLs), supported in both UFS and NFS, provide greater flexibility in managing file permissions than traditional UNIX file protection. The traditional UNIX file protection provides read, write, and execute permissions for three user classes: owner, group, and other.

Using ACLs allows you to define file permissions for the owner, owner's group, others, specific users and groups, and default permissions for each of those categories. For example, you can set up an ACL that defines read permission to a group of users and write permission to only one user in the group. You could not do this with standard UNIX file permissions.

The `setfacl(1)` command sets, adds, modifies, and deletes ACL entries, and the `getfacl(1)` command displays ACL entries.

See *System Administration Guide* for more information about using ACLs.

Automated Security Enhancement Tool (ASET)

The Automated Security Enhancement Tool (ASET), available as a separate option with SunOS release 4.x systems, is included with the Solaris 2.6 operating environment. ASET enables you to specify an overall system security level (low, medium, or high) and automatically maintain systems at those levels. This tool can be set up to run on a server and all its clients or on individual clients.

ASET performs these tasks:

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

System Administration Guide describes ASET setup and monitoring in detail.

Security Options

Currently available bundled security options are Kerberos security, the SunSHIELD™ package, and Pluggable Authentication Module (PAM).

Kerberos 4.0 Security

The Solaris 2.6 operating environment includes support for Kerberos V4 authentication for secure RPC. (Kerberos source code and administrative utilities are available from MIT.) Included in this release are:

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

System Administration Guide describes the client-side utilities, included in the release. *NFS Administration Guide* describes the use of Kerberos with the NFS application.

SunSHIELD Package

The Solaris 2.6 release includes the SunSHIELD Basic Security Module (BSM) package. This product provides the security features defined as C2 in the Trusted Computer System Evaluation Criteria (TCSEC). The features provided by the BSM are a security auditing subsystem and a device allocation mechanism. C2 discretionary access control and identification and authentication features are provided in the operating system.

The administration of BSM is included in *SunSHIELD Basic Security Module Guide*.

PAM

The Pluggable Authentication Module (PAM) framework enables new authentication technologies to be “plugged-in” without changing commands, such as `login`, `ftp`, `telnet` and so on. The framework enables a system administrator to choose any combination of services to provide authentication. Mechanisms for account, session, and password management can also be “plugged-in” using this framework.

System Administration Guide describes the administration of PAM.

User Environment Administration

This chapter describes differences in tasks you may perform to set up the local user environment after installing the Solaris 2.6 software.

- “Selecting a Default Shell” on page 45
- “Customizing User Environments” on page 47
- “Window Systems ” on page 48
- “User and Group Administration ” on page 49
- “ Using Mail” on page 50
- “Using Document Tools” on page 51
- “Man Page Organization Differences” on page 51

Selecting a Default Shell

The login shell is the command interpreter that runs when you are logged in. The Solaris 2.6 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. 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

To change from one shell to another, use one of the following methods:

- *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 a windows environment, use Admintool. See *Solaris Advanced User's Guide* for information.

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

Customizing User Environments

This section describes how to determine which initialization files you can edit to customize the local environment based on your choice of login shell, and where to find them in the SunOS release 5.6 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 the SunOS release 4.x software. The template file locations are shown in . 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	/etc/skel/local.profile
C	/etc/skel/local.login /etc/skel/local.cshrc
Korn	/etc/skel/local.profile

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

Using the SunOS 4.x Work Environment With the Solaris Software

The SunOS release 5.6 software can use the old SunOS release 4.x system files and initialization files such as `.login`, `.cshrc`, and `.profile` to re-create the look and feel of the SunOS release 4.x work environment. 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, explains how to re-create the SunOS release 4.x environment within the Solaris 2.6 operating environment.

Window Systems

The Common Desktop Environment (CDE) is the default Solaris 2.6 windowing environment and offers a simple and intuitive interface. See Chapter 14, for more information about CDE.

The OpenWindows 3.1 software can also be used as your preferred desktop with the Solaris 2.6 environment. 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 the SunOS release 5.6 software and its usage is correct.

SunView™ software is not part of the Solaris 2.6 operating environment. 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 the OpenWindows 3.1 environment
- The applications that are not compatible between OpenWindows Version 2.0 and 3.1 platforms
- Guidelines for modifying incompatible applications

User and Group Administration

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

User and Group Administration Choices

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

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.

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`, and so on).
4. Add the new user to the naming service (NIS).

In the Solaris 2.6 operating environment, there are three ways to add (and maintain) user accounts:

- Use `Admintool` – This is the most straightforward method to use if the system is running the OpenWindows environment.
- Use command-line interfaces (`useradd`, `usermod`, and `userdel`) – Use this method if you don't want to use `Admintool`.
- Manually edit files (similar to the SunOS release 4.x procedure with a few exceptions)

Note - Because the SunOS release 5.6 software uses a shadow password file, simply editing the `/etc/passwd` file is no longer sufficient. You should not attempt this method unless you have ample experience with this type of administration.

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

Using Mail

The SunOS release 4.x mail programs are different in the Solaris 2.6 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 2.6 operating environment's version of `mail`. Additionally, some useful mail facilities are included for compatibility.

In the Solaris 2.6 operating environment, there are three programs for sending and retrieving your mail. All three are backward compatible and can be used to read your SunOS release 4.x mail. They are:

- `mailtool`, the OpenWindows interface for the mail program. New Solaris 2.6 `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 2.6 mail reading program. It is an enhanced version of SunOS release 4.x `/usr/ucb/mail`. In the Solaris 2.6 operating environment, `/usr/ucb/mail` is a link to `/usr/bin/mailx`. `mailx` offers message headers that enable 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 2.6 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 2.6 mail programs, see *Mail Administration Guide*.

Using Document Tools

This section outlines the main differences in using document tools between SunOS release 4.x and the Solaris 2.6 operating environment.

- The Solaris 2.6 operating environment provides a set of PostScript filters and device-independent fonts. However, most SunOS release 4.x TranScript filters have SunOS release 5.6 equivalents while a few less common ones do not. In SunOS release 5.6 systems, there is no T_EX filter, no `pscat` (C/A/T) filter, and no raster image filter.
- The Solaris 2.6 operating environment provides device-independent `troff`, with the following features: SunOS release 4.x `troff` input files work with Solaris 2.6 `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 page organization has changed to be compatible with SVR4 organization. As a result, some sections have been renamed. For example, `man(8)` is now `man(1M)`.

Table 6-4 shows SunOS release 5.6 man page directories.

TABLE 6-4 SunOS release 5.6 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 3S - Standard I/O functions 3T - Threads library functions 3X - Miscellaneous library functions
man4	File formats	4B - SunOS/BSD compatibility file formats

TABLE 6-4 SunOS release 5.6 man Page Directories (continued)

/man Directory	Contents	Suffixes
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 in the SunOS release 4.x software, which searched the individual man directories according to a predetermined order, the SunOS release 5.6 software 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/
```

```
sunos5.6% ls /usr/share/man
man.cf manlf/ 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

The SunOS release 4.x man page table of contents and keyword database is called `whatis`. In the SunOS release 5.6 software, 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
```

```
sunos5.6% man -k tset
reset tset (1b) - establish or restore terminal characteristics
tset tset (1b) - establish or restore terminal characteristics
```

Using the man Command

Table 6-5 shows that SunOS release 5.6 version of the `man` command has additional search options.

TABLE 6-5 New man Command Options

Option	Description
-a	Displays all man pages that match <i>file name</i> . The pages are displayed sequentially in the order they are found.
-l	Lists all man pages that match file name. You can use the output of this command to specify a section number with the -s option.
-s <i>section-number</i>	Searches <i>section-number</i> for <i>file name</i> . In the SunOS release 4.x software, the man command accepted the section number as an option; in this release, the section number must be preceded by -s.
-F	Forces the man 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 5.6 man command.

Device Administration

This chapter explains SunOS release 5.6 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.

- “Device Naming Conventions” on page 57
- “Obtaining Disk Information” on page 59
- “Adding Devices to the System” on page 61
- “Using Volume Management ” on page 61

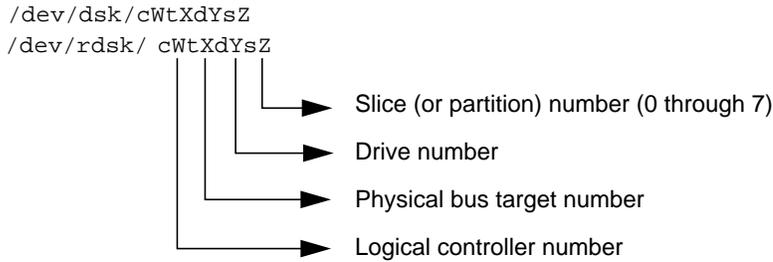
Device Naming Conventions

Device naming conventions have changed between the SunOS release 4.x and SunOS release 5.6 platforms. 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/rdisk`.

SunOS release 5.6 commands that take device names as arguments must use the SunOS release 5.6 device naming conventions. However, you can still use and recognize the SunOS release 4.x device names if you install the SunOS/BSD Source Compatibility Package. See *Source Compatibility Guide* for additional information.

Convention for Disks

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



Note - Most SCSI disks have embedded controllers. This means that the drive number will always be 0 but the target number varies. For example, if an external disk drive has its rear switch set to 2, the device name for the first slice is `/dev/dsk/c0t2d0s0`, 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 the SunOS 4.1.x software, SCSI target 3 was called `sd0`, but it is now properly named `c0t3d0`. SCSI target 0 was called `sd3`, but it is now named `c0t0d0`. Other SCSI disk names translate normally. For example, in the SunOS release 5.6 software, `sd2a` is `c0t2d0s0` and `sd2b` is `c0t2d0s1`.

Convention for Tape Drives

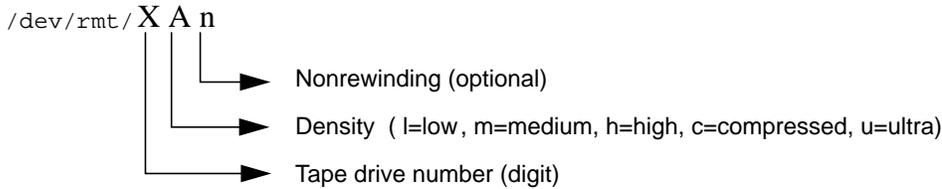


Table 7-1 provides some examples that compare the SunOS release 4.x and SunOS release 5.6 device naming conventions.

TABLE 7-1 SunOS release 4.x and SunOS release 5.6 Device Names

Device Description	SunOS release 4.x Device Name	SunOS release 5.6 Device Name
Disk devices	/dev/sd0g	/dev/dsk/c0t3d0s6
	/dev/rsd3b	/dev/rdisk/c0t0d0s1
	/dev/rsd3a	/dev/rdisk/c0t0d0s0
Magnetic tape devices	/dev/nrmt8	/dev/rmt/8hn
	/dev/rst0	/dev/rmt/0
CD-ROM device	/dev/sr0	/dev/dsk/c0t6d0s2

Obtaining Disk Information

The commands that report disk information in the SunOS release 5.6 software have changed. `df(1M)` and `du(1M)` are still available, but have changed. `dinfo(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 5.6 command differs significantly from that used in the SunOS release 4.x version (see Appendix A, for more information). For more information on VFS, see “Virtual File System Architecture” on page 77.

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 5.6 version produces a full listing with totals.

Finally, use the SunOS release 5.6 device naming conventions when specifying special device names to this command. See “Device Naming Conventions” on page 57 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.

dinfo Command

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

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

The following screen shows sample output for the SunOS release 5.6 `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    Sector Count    Last Sector    Mount
Directory
   0          0    00         0          32724    32723    /
   1          0    00        32724       65448    98171
```

(Continued)

2	0	00	0	406296	406295	
6	0	00	98172	308124	406295	/usr

devinfo Command

The SunOS release 4.x version of `devinfo` is incompatible with the SunOS release 5.6 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 if you do, you will be asked to provide other boot parameters, including what to boot and where the root file system is.

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

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

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.

The OpenWindows and CDE File Manager applications have 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 CD and diskette file systems when removable media are inserted into the devices. Any CD or diskette file system will be automatically mounted in the locations described in Table 7-2.

TABLE 7-2 Location of CD-ROM and 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 locations described in Table 7-3.

TABLE 7-3 Location of CD-ROM and 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 and on using diskettes and CDs, see *System Administration Guide*.

Man 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` and `/dev/rdisk/c0t6d0s0`; and these diskette paths: `/dev/diskette` and `/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 described in Table 7-4.

TABLE 7-4 Volume Management Commands

Command	Description
<code>rmmount(1)</code>	Removable media mounter. Used by <code>vold</code> to automatically mount <code>/cdrom</code> and <code>/floppy</code> when a CD or diskette is installed.
<code>volcancel(1)</code>	Cancels a user's request to access a particular CD-ROM or diskette file system
<code>volcheck(1)</code>	Checks drive for installed media. By default, checks drive pointed to by <code>/dev/diskette</code> .
<code>volmissing(1)</code>	Notifies user when an attempt is made to access a CD or diskette that is no longer in the drive
<code>vold(1)</code>	Volume Management daemon, controlled by <code>/etc/vold.conf</code>

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

Startup and Shutdown

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

- “Booting” on page 65
- “Using the `init` Command” on page 68
- “Shutting Down” on page 70

See *System Administration Guide* 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 2.6 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 boot time.
- The boot block can read UNIX file systems, eliminating boot errors when the boot program moves.
- Third-party bootable devices are supported.
- Secondary boot programs, `ufsboot` and `inetboot`, have been modified to read CacheFS file systems. This new booting capability enables Solstice AutoClient™ systems to boot more quickly and with less impact on network resources.

- The SunOS release 4.x `fastboot` command is available only on Solaris 2.6 systems that have the SunOS/BSD Source Compatibility Package installed. The `fastboot` command is obsolete in Solaris 2.6 systems because file systems are only checked if the file system state is identified as not clean.
- The SunOS release 4.x `halt` and `reboot` commands have `shutdown(1M)` and `init(1M)` equivalents in the SunOS release 5.6 software.

In the Solaris 2.6 operating environment, the `shutdown` and `init` commands are the preferred way to halt, shut down, or reboot your system. While the `reboot` command is available in the Solaris 2.6 operating environment, it brings the system down quickly without shutting down services in an orderly way. Table 8-1 shows the SunOS release 5.6 commands that replace SunOS release 4.x commands.

TABLE 8-1 SunOS release 5.6 Replacements for `reboot` and `fastboot`

SunOS release 4.x Command	SunOS release 5.6 Command Replacement
<code>reboot</code>	<code>shutdown -i -6, init 6</code>
<code>fastboot</code>	<code>boot, init 6</code>

boot Command Changes

The SunOS release 5.6 software 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. The messages are always stored in the `/var/adm/messages` file.
- Type `boot -a` when you want to be prompted for the name of an alternate kernel, `/etc/system` file, or path name for kernel module directories.

Booting From the PROM

Be aware of these changes when booting from PROM:

- The PROM loads `bootblk` 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 the SunOS release 4.x software, you need to use `installboot(1M)` to install boot blocks on a partition to be used for booting.
- `bootblk` opens the boot device and, using the file system you specify, finds and loads `ufsboot`.
- The boot PROM loads the kernel, `/kernel/genunix`, after `ufsboot` is loaded into memory. SunOS release 4.x systems used `/vmunix`; however, in the SunOS release 5.6 software the `/kernel` directory contains all platform-independent kernel modules, including `unix`, needed to boot the system.
- The kernel, in turn, loads other drivers, such as `esp`, 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 5.6 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 5.6	Feature
<code>bootsd</code>	<code>bootblk</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/genunix</code>	Bootable kernel image
<code>boot.sun4c.sunos.4.1</code>	<code>inetboot</code>	Mounts and copies <code>unix</code> from network

TABLE 8-2 Summary of Booting Differences (continued)

SunOS release 4.x	SunOS release 5.6	Feature
<code>rc.boot rc.single</code>	<code>/etc/rcS</code>	Mounts <code>/usr</code> and checks file systems
<code>rc.local</code>	<code>/etc/rc2 /etc/rc3</code>	System config scripts
<code>/etc/config</code>	<code>modload /etc/system</code>	Customizes system kernel, loads modules as needed
Prom monitor, single user, multiuser	Run states 0 - 6, and S	System run levels
<code>/dev/sd1g</code>	<code>/dev/dsk/c0t1d0s6</code>	More descriptive logical device names. See "Device Naming Conventions" on page 57.
MAKEDEV	<code>boot -r,</code> <code>add_drv</code>	Makes device nodes

Using the `init` Command

The `init(1M)` command replaces the SunOS release 4.x `fasthalt` command in the SunOS release 5.6 software. 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 5.6 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 5.6 `init` command uses a different script for each run level instead of grouping all 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.

System Administration Guide describes this command in detail.

Changing System Run Levels

The SunOS release 5.6 `init` command enables you to control the run level (initialization state) of your system and move easily between various modes of operation. The SunOS release 5.6 `/sbin/rc` scripts 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.

The SunOS release 4.x software had three run levels: prom monitor, single user, and multiuser. These correspond to run levels 0, 1, and 3 in the SunOS release 5.6 software.

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

TABLE 8-3 SunOS release 5.6 System Initialization Run Levels

Run Level	Default SunOS release 5.6 Function
0	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.
1	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.
2	Normal multiuser operation without NFS systems exported. Sets the <code>timezone</code> 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.
3	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.
4	Alternative multiuser state (not used).

TABLE 8-3 SunOS release 5.6 System Initialization Run Levels (continued)

Run Level	Default SunOS release 5.6 Function
5	Software reboot, prompts for a boot device other than the EEPROM default.
r	Reboot. Kills all active processes. Unmounts the file systems and reboots to the <i>initdefault</i> entries in <i>/etc/inittab</i> .
S,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 5.6 systems that have the SunOS /BSD Source Compatibility Package installed.
- The SunOS release 4.x `halt` and `reboot` commands have `shutdown` and `init` equivalents.

See *System Administration Guide* for detailed descriptions of shutdown procedures. .

In the SunOS release 5.6 software, the `shutdown` command is the preferred way to halt or shut down a system. `shutdown` and `init` use `rc` scripts to kill running processes. While the `halt` command is available in the SunOS release 5.6 software, it stops the system quickly without shutting down services in an orderly way. Table 8-4 shows the SunOS release 5.6 commands that replace those in the SunOS release 4.x system.

TABLE 8-4 SunOS release 5.6 Replacements for `halt` and `fasthalt`

SunOS release 4.x Command	SunOS release 5.6 Command Replacement
<code>halt</code>	<code>shutdown -i 0, init 0</code>
<code>fasthalt</code>	<code>shutdown -i 0, 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 for information about the run-level numbers.

Changes to the `shutdown` Command

The SunOS release 5.6 `shutdown` command includes only the options in Table 8-5. This command and its options are described in *System Administration Guide*.

TABLE 8-5 SunOS release 5.6 `shutdown` Command Options

Option	Description
<code>-g</code>	Selects “grace” period before <code>shutdown</code> begins.
<code>-i [init state]</code>	Specifies an initial run level (see Table 8-3).
<code>-y</code>	Runs <code>shutdown</code> without asking confirmation questions. Assumes a “yes” response to all questions.
<code>-message</code>	Specifies user-supported message. If more than one word, use quotes around the message.

By default, the SunOS release 5.6 `shutdown` command asks you to confirm before an actual `shutdown` begins. You can use the `-y` option to run it without operator intervention.

The `shutdown` options are available only in BSD source compatibility mode on Solaris 2.6 systems.

See Appendix A, for a summary of changes. See `shutdown(1M)` 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 2.6 systems. The file-system checking features of these commands are not appropriate to a Solaris 2.6 system.

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 5.6 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
- “File System Changes” on page 73
- “Default File Systems and Directories” on page 75
- “Virtual File System Architecture” on page 77
- “Directory and File Changes” on page 82
- “Using File System Administration Commands” on page 90

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

File System Changes

SunOS release 5.6 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.
- The `/etc/vfstab` tab file replaces `/etc/fstab`.
- The `/etc/lp` directory replaces `/etc/printcap`.
- The SunOS release 5.6 `/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 5.6 `/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.
- 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 5.6 software.

The added pseudo file systems 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)` man 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 5.6 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 `voldfs(7)` man page.

Default File Systems and Directories

The SunOS release 5.6 file system is hierarchical. Figure 9-1 graphically depicts SunOS release 5.6 default directories and file systems (indicated by dotted lines). Subdirectories shown are just a sample of what the directory or file system actually holds. Table 9-1 gives a brief description of each.

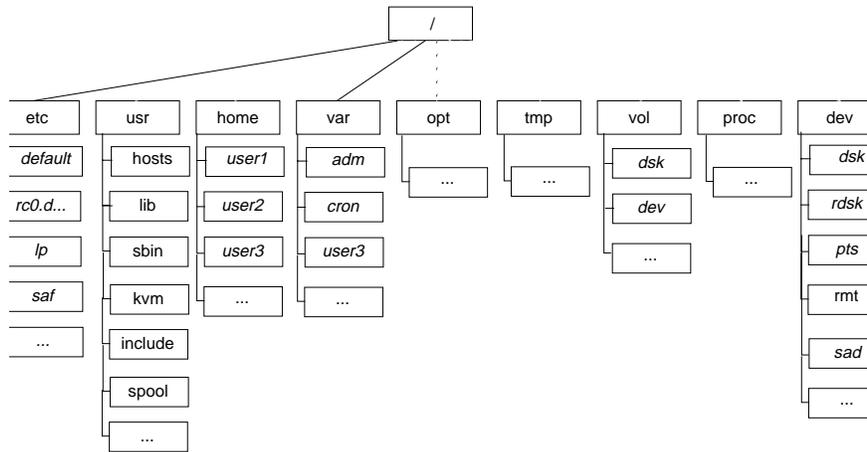


Figure 9-1 Solaris 2.6 Default File Systems and Directory Hierarchy

The Solaris 2.6 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 the default file systems and directories with a brief description.

TABLE 9-1 Solaris 2.6 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 UFS 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

The SunOS release 5.6 software 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, with each its own set of commands for file system management. Learning all the variations can be confusing and difficult. The SunOS release 5.6 software addresses this issue with a set of generic commands for file system management. These commands are a part of 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 the SunOS release 4.x software are also included in the SunOS release 5.6 software. There is one exception: The translucent file system (TFS) type has been withdrawn from the SunOS release 5.6 software. Table 9-2 summarizes file-system type availability in the SunOS release 4.x and SunOS release 5.6 environment.

TABLE 9-2 Summary of File System Types

Category	Name	Description	SunOS release 4.x	SunOS release 5.6
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

TABLE 9-2 Summary of File System Types *(continued)*

Category	Name	Description	SunOS release 4.x	SunOS release 5.6
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
	FIFOFS	FIFO/Pipe file system	No	Yes
	NAMEFS	Name file system	No	Yes
	SWAPFS	Swap file system	No	Yes
	CACHEFS	Cache file system	No	Yes

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

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

In the SunOS release 5.6 software SWAPFS is the 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 to `sd0b` and diskless systems get their swap files from the `bootparam` server. The SunOS release 5.6 software 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 the SunOS release 5.6 software.

TABLE 9-3 Not Supported SVR4 File System Types

Name	Description
<code>BFS</code>	Boot file system
<code>S5</code>	System V file system
<code>xnamefs</code>	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

TABLE 9-4 Generic File-System Administrative Commands *(continued)*

Command	Description
<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
<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; 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, the argument 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(4)`) 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 the SunOS release 5.6 software, 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`.

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 the SunOS release 4.x and SunOS release 5.6 environment.

`/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

TABLE 9-6 Additions to the /dev Directory (continued)

Subdirectory	Description
/dev/pts	Contains pseudo terminal (pty) slave devices
/dev/rmt	Contains raw tape devices
/dev/sad	Contains entry points for the STREAMS Administrative Driver
/dev/term	Contains terminal devices

/etc Directory

The /etc directory contains system configuration information. Several files and subdirectories have been added, removed, or changed.

- 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 the SunOS release 5.6 software. They are replaced by the scripts shown in Table 9-7, which are run by their corresponding run control files. Table 9-8 describes the subdirectories that have been added to the SunOS release 5.6 /etc directory.

TABLE 9-7 Initialization Scripts and Their Run Control Files

Scripts	Run Control Files
/etc/rc0.d	/sbin/rc0
/etc/rc1.d	/sbin/rc1
/etc/rc2.d	/sbin/rc2
/etc/rc3.d	/sbin/rc3

TABLE 9-7 Initialization Scripts and Their Run Control Files *(continued)*

Scripts	Run Control Files
/etc/rc4.d	/sbin/rc4
/etc/rc5.d	/sbin/rc5
/etc/rc6.d	/sbin/rc6
/etc/rcS.d	/sbin/rcS

TABLE 9-8 Additions to the /etc Directory

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

/etc/vfstab File

In the SunOS release 5.6 software, 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 79.

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 systems 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
<code>device to mount</code>	<p>The entry in this field may be any of the following:</p> <p>The block special device for local UFS 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 NFS 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>hsfs</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>
<code>device to fsck</code>	<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>
<code>mount point</code>	<p>The default mount-point directory (for example, <code>/usr</code> for <code>/dev/dsk/c0t0d0s6</code>).</p>

TABLE 9-9 /etc/vfstab File Field Names and Content (continued)

Field Name	Content
FS type	The type of file system identified by the device to mount field.
fsck pass	The pass number used by fsck 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-UFS file systems with a 0 fsck pass are checked. For UFS file systems only, when the field contains a 0, the file system is not checked. When fsck is run on multiple UFS file systems that have fsck pass values greater than 1 and the preen option (-o p) is used, fsck 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 5.6 system software, the fsck pass field does not explicitly specify the order in which file systems are checked.
automount?	yes or no for whether the file system should be automatically mounted by mountall when the system is booted. An auto in the fourth column of your SunOS release 4.x /etc/fstab would translate to a "yes" in this column; a noauto, a "no." Note that this field has nothing to do with the automount program.
mount options	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 mount(1M) man page for a list of the available options.

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

/etc/shadow File

The SunOS release 5.6 software contains an /etc/shadow file, which 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 5.6 /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 and “Changing System Run Levels” on page 69 for a description of the scripts.

/usr Directory

The SunOS release 5.6 /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 5.6 /usr directory.

TABLE 9-10 Additions to the /usr Directory

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

Table 9-11 shows files that were in the SunOS release 4.x /usr directory but have been moved in the SunOS release 5.6 software.

TABLE 9-11 Files Changed in the /usr Directory

SunOS release 4.x Location	SunOS release 5.6 Location
/usr/5bin	/usr/bin
/usr/5include	/usr/include
/usr/5lib	/usr/lib
/usr/etc	/usr/sbin
/usr/old	Contents removed
/usr/xpg2bin	/usr/bin

TABLE 9-11 Files Changed in the `/usr` Directory (continued)

SunOS release 4.x Location	SunOS release 5.6 Location
<code>/usr/xpg2lib</code>	<code>/usr/lib</code>
<code>/usr/xpg2include</code>	<code>/usr/include</code>

Appendix E, 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.

- 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`.

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

`/kernel` Directory

The SunOS release 5.6 `/kernel` directory contains the operating system kernel and kernel-level object modules that were in the SunOS release 4.x `/sys` directory. Table 9-12 describes the subdirectories that have been added to the `/kernel` directory.

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> , and so on
<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 5.6 `/opt` directory contains optional add-on application software packages. These packages were installed in the SunOS release 4.x `/usr` directory.

`/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 in the SunOS release 5.6 software 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 5.6 system, see *System Administration Guide* 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.

`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 in the maps which servers are preferred for each resource 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.**

Specifying the mount point is usually easier. The rest of the information is read from `/etc/vfstab`.

4. **Become root and type the `mount` command, specifying either the mount point or the block device.**

Specifying the mount point is usually easier. The rest of the information is read from `/etc/vfstab`.

```
# mount mount-point
```

The file system is now mounted.

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

Changes to the `mount` Command

Some of the names and forms of the `mount` commands are different, as listed in Table 9-13.

TABLE 9-13 `mount` Command Differences

SunOS Release 4.x	SunOS release 5.6
<code>mount</code>	<code>mount</code>
<code>mount -a</code>	<code>mountall</code>
<code>umount</code>	<code>umount</code>
<code>umount -a</code>	<code>umountall</code>
<code>exportfs</code>	<code>share</code>
<code>exportfs -u</code>	<code>unshare</code>

TABLE 9-13 mount Command Differences (continued)

SunOS Release 4.x	SunOS release 5.6
<code>showmount -a</code>	<code>dfmounts</code>
<code>showmount -e</code>	<code>dfshares</code>

See Appendix A, 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 are 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 61 for more information.

Specifying File Systems in the `/etc/vfstab` File

In the SunOS release 5.6 system, you need to list file systems that you want mounted at system startup in your `/etc/vfstab`, instead of in the `/etc/fstab` file. The format of `/etc/vfstab` differs from that of `/etc/fstab`. For a discussion of the `/etc/vfstab` file, see “`/etc/vfstab` File ” on page 84.

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 5.6 <code>df</code> command. The SunOS release 5.6 <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 5.6 <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 <code>-n cpio-device</code> SunOS release 4.x option is not available in the SunOS release 5.6 command. Write the current file on device in <code>cpio -c</code> format.

Sharing File Systems

File systems were “exported” in the SunOS release 4.x software 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 the SunOS release 5.6 software, 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.

`newfs` Command

The SunOS release 5.6 `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 5.6 device naming conventions (see “Device Naming Conventions” on page 57).

`mkfs` Command

The SunOS release 5.6 `mkfs` command differs significantly from the SunOS release 4.x version of the command. The SunOS release 5.6 version provides for different file system types, and its command syntax is entirely different (see “Generic File System Commands” on page 79). Like `newfs`, `mkfs` accepts only names conforming to the SunOS release 5.6 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 `mkfs(1)` man pages for additional details.

Checking File Systems

The SunOS release 5.6 `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 79).

In addition, `fsck` accepts only names conforming to the SunOS release 5.6 device naming conventions. For more information, see “Device Naming Conventions” on page 57.

The `fsck` command performs faster consistency checks at mount time. In addition, the SunOS release 5.6 software does not require you to reboot the system after running `fsck` on the root and `/usr` file systems. This results in faster system startup compared to previous SunOS releases. The `fsck -m` command enables you to skip checking for file systems that are clean. See `fsck(1m)` for additional details.

Backing Up and Restoring Files

This section discusses the changes to backup and restore commands and SunOS release 5.6 and describes how to use the `ufsdump`, `ufsrestore`, `dd`, `tar`, and `cpio` commands.

The SunOS release 4.x software 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 5.6 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 5.6 system with `ufsrestore`.

The SunOS release 5.6 software has two additional utilities for copying file systems: `volcopy(1M)` and `labelit(1M)`.

`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 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 you no longer have 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 the SunOS release 5.6 software. In the SunOS release 4.x software, 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 5.6 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 5.6 system, use the device naming convention to identify the tape drive; otherwise, use the SunOS release 4.x convention.

ufsrestore Command

The `ufsrestore` command in the SunOS release 5.6 software is similar to the `restore` command in the SunOS release 4.x software. You will be able to restore backups made with the SunOS release 4.x `dump` command with one exception: you cannot restore multivolume backups from diskette. If you have backup scripts that invoke `restore`, change them to invoke `ufsrestore` instead.

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 5.6 version, `-w` denotes a unit of 2 bytes. In addition, the SunOS release 5.6 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 the SunOS release 5.6 software, 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 5.6 `cpio` command supports the SunOS release 4.x options and command syntax. `cpio` has been expanded to include many new options, as listed in Table 9-17.

TABLE 9-17 Additional `cpio` Options

Option	Command Available With Option	Description
<code>-A</code>	<code>cpio -o</code>	Appends files to an archive.
<code>-k</code>	<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.
<code>-L</code>	<code>cpio -o</code> or <code>cpio -p</code>	Follows symbolic links.
<code>-v</code>	<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.
<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: bar (read only), crc, CRC, odc, tar, TAR, ustar, or USTAR.
<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>	Defines a message to use when switching media.

TABLE 9-17 Additional `cpio` Options (continued)

<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> .

Note - `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 2.6 Server to Support SunOS Release 4.x Diskless/Dataless Clients

This chapter outlines how to set up a Solaris 2.6 system as a server for SunOS release 4.x diskless/dataless clients by using the `discover4x`, `install4x`, and `convert4x` programs.

Make sure you have read Chapter 3, if you are setting up a Solaris 2.6 server for SunOS release 4.x clients on a Solaris 2.6 network.

- “Adding SunOS Release 4.x Support to a Solaris 2.6 Server” on page 101
- “Running `discover4x`” on page 102
- “Setting Up the CD-ROM Drive for `install4x` ” on page 103
- “Running `install4x` ” on page 105
- “Running `convert4x`” on page 108

Adding SunOS Release 4.x Support to a Solaris 2.6 Server

This section explains how to prepare a Solaris 2.6 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.

Some sites will need to continue using SunOS release 4.x clients after the server has been upgraded to Solaris 2.6 software. For instance, Sun-3TM systems cannot run Solaris 2.2 or later software and must continue to use the SunOS release 4.x software.

When a SunOS release 4.x `/export` partition is set up on a server running Solaris 2.6 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 2.6 operating environment.

The multiple OS operation package is called SUNWhinst and includes three programs that you will need to run to set up a SunOS release 4.x `/export` directory on a Solaris 2.6 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 2.6 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 2.6 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 2.6 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 2.6 operating environment.
- `convert4x` – This program updates the Solaris 2.6 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 2.6 server.

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

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

Running `discover4x`

`discover4x` analyzes the support that remains for SunOS release 4.x clients after the server has migrated to the Solaris 2.6 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 2.6 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 2.6 operating environment using one of the three procedures listed in the following section.

- If the system has a local CD-ROM drive, see “Using a Local CD-ROM Drive” on page 103
- If the system will use a remote CD-ROM drive on a system running the Solaris 2.6 operating environment, see “Using a Remote CD-ROM Drive (Solaris 2.6 Software)” on page 104
- If the system will use a remote CD-ROM drive on a system running the SunOS release 4.x software, see “Using a Remote CD-ROM Drive (SunOS Release 4.x Software)” on page 104

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 2.6 Software)

If `install4x` is to use a CD-ROM drive on a remote system running the Solaris 2.6 operating environment, after you install the CD into the drive, Volume Management automatically mounts the CD directory on `/cdrom/volume1/s0`. Then type 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 Software)

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

```
# mkdir /cdrom
# mount -t hsfs -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 command.

```
# /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 command.

```
# /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
                                Selected      Loaded
[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 module
or abort without loading any modules
  [Q] Quit without loading

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

Type any bracketed letter to select that function.
```

(Continued)

Type ? for help.

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
+[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 OpenWin_Users.25936K      [z] Security.....312K
[m] C OpenWin_Demo...4288K     [A] OpenWindows_Progr..10200K
[n] C OpenWin_Fonts..7840K

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

Select [a-A] or a Quick-Pick Option:      +-----+
[1] All Req'd Modules [4] All Opt Moduls | Disk Usage: |
[2] All Desr'ble Mod [5] All Modules      | 0K Selected  |
[3] All Common Modules                    | 53634K Free  |
or [D] (done) to return to the main scrn +-----+

```

Packages already loaded are shown on the Module Selection screen with a plus sign (+) 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 asterisks (**) 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.sun4c.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           +-----+
                                     | Disk Usage: |
or abort without loading any modules: |   0K Selected |
                                     | 53634K Free  |
                                     +-----+

[Q] Quit without loading

Type any bracketed letter to select that function.
Type ? for help.

```

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 ...

```

(Continued)

```
Press any key to continue:
```

Running `convert4x`

`convert4x` updates the Solaris 2.6 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

Before running `convert4x`, 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.

As Superuser, run `convert4x` by typing the following command.

```
# /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 add them again using Solstice Host Manager.

Managing Printers, Terminals, and Modems

This chapter describes how to manage printing, and the differences in print commands, in the Solaris 2.6 environment. It also describes serial port management (which enables terminal and modem connections) through Admintool or the Service Access Facility (SAF),

- “Summary of Printing Differences” on page 109
- “Using Printer Commands” on page 110
- “Terminal and Modem Management” on page 113
- “Service Access Facility (SAF)” on page 113

Printing

This section describes how to set up and administer printers after you install Solaris 2.6 software. This chapter also describes the changes to printer commands that have taken place between the SunOS release 4.x and the Solaris 2.6 release environments.

Summary of Printing Differences

The SunOS release 5.6 LP print service replaces the SunOS release 4.x printing facilities, which were provided by the `lpd` daemon and `lpr`, `lpq`, `lprm`, and `lpc` commands. Admintool enables you to 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 5.6 printers.

For detailed information about Admintool and the command-line interface to the LP service, see *System Administration Guide*.

The services provided by the `/etc/printcap` file in the SunOS release 4.x software are handled in the Solaris 2.6 operating environment by the `terminfo` database and by the files in the `/etc/lp` directory.

Print Commands and the Compatibility Package

You can still use many SunOS release 4.x print commands if the system is running the SunOS/BSD Source Compatibility Package. Compatibility mode uses SunOS release 4.x command names as an interface to underlying Solaris 2.6 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 2.6 system, the commands create message files that are handled by the SunOS release 5.6 LP print service scheduler.

Solaris 2.6 printing provides additional capabilities not available in SunOS release 4.x systems. These capabilities enable you to control forms, print wheels, and interface programs, and to set up network print services.

Using Printer Commands

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

TABLE 11-1 User Print Command Equivalents

SunOS release 4.x	SunOS release 5.6	Function
<code>lpr filename</code>	<code>lp filename</code>	Print a file to the default printer
<code>lpr -P printer 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

TABLE 11-1 User Print Command Equivalents *(continued)*

SunOS release 4.x	SunOS release 5.6	Function
Check /etc/printcap	lpstat -d	Determine which is the default printer
Check /etc/printcap	lpstat -a	Determine which printers are available
lprm <i>jobnumber</i>	cancel <i>jobid</i>	Cancel a print job on the default printer

Using SunOS Release 5.6 Printer Administration Commands

This section describes differences between printer setup and administration on SunOS release 4.x and Solaris 2.6 systems. All the underlying system services described are available only in the Solaris 2.6 operating environment. The SunOS release 4.x counterparts are not available even in compatibility mode.

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

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

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

SunOS release 4.x	SunOS release 5.6	Function
lpc	lpadmin	Control line printer functions
/etc/printcap	terminfo database and <code>/etc/lp/printers/ printername/*</code>	File that defines printer functions

TABLE 11-2 Printer Administration, Setup, and File Equivalents *(continued)*

SunOS release 4.x	SunOS release 5.6	Function
/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 queuing to a printer

Printing troff

In the SunOS release 4.x software, you need the following command to send a troff file to the default printer.

```
% troff filename
```

In the Solaris 2.6 operating environment, you must specify that you want the file printed by piping (|) the output to the lp command. Table 11-3 shows the SunOS release 5.6 troff commands.

TABLE 11-3 SunOS release 5.6 troff Commands

SunOS release 5.6 Command	Function
troff <i>file</i> /usr/lib/lp/postscript/dpost lp	Sends to default printer that supports troff jobs
troff <i>file</i> /usr/lib/lp/postscript/dpost lp -d <i>printer</i>	Sends to a particular printer
troff <i>file</i> lp-Ttroff	Sends to any printer that supports troff jobs

Serial Port Management

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

System Administration Guide describes the details of Solaris 2.6 setup and installation procedures for serial devices.

Terminal and Modem Management

Admintool is a tool that readily enables you to set up and modify serial port software for terminals and modems.

Admintool provides:

- 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.

Controlling Port Monitors

SAF's common interface helps control services called port monitors. 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 2.6 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.

SAF Functions and Related Programs

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

The `sacadm` command enables you to 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 for customizing port monitors.

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.

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. The functions and associated programs are:

- Overall administration – `sacadm`
- Port Monitor Service Administrator – `pmadm`

- Service Access Control - `sac`
- Port monitors - `ttymon` and `listen`
- Services - logins, remote procedures

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.

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

Network Service Administration

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

- “Changes to TCP/IP” on page 117
- “Changes to NFS” on page 118
- “PPP ” on page 118
- “UUCP” on page 119

Changes to TCP/IP

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

The NIS+ maps administered by AdminTool include:

- Hosts
- Services
- RPC
- Ethers

When you are ready to configure SunOS release 5.6 TCP/IP facilities, see *TCP/IP and Data Communications Administration Guide* for information about setting up TCP/IP.

Changes to NFS

The Solaris 2.6 operating environment simplifies resource sharing with a new set of commands and files to administer NFS resources. In specific, `exportfs` and `/etc/exports` has been replaced by `share`, `shareall`, and `/etc/dfs/dfstab`. This new command set was designed to allow for future distributed file system types.

Several of the daemons associated with NFS have been renamed. `rpc.statd`, `rpc.lockd`, and `rpc.mountd` are now simply called `statd`, `lockd`, and `mountd`.

Unlike the SunOS release 4.x environment, there are no client side block I/O daemons (biods) in the Solaris 2.6 release. They have been superceded by kernel threads. Also, the NFS daemon, `nfsd`, has been altered so that it does not spawn multiple copies to handle concurrent requests.

Other features included in this release:

- NFS over TCP
- NFS Version 3
- Improved NFS Lock Manager
- Support for Access Control Lists (ACLs)
- WebNFS
- NFS Client Failover
- Kerberos support for NFS file systems
- NFS Large File Support

All these features are described in *NFS Administration Guide*.

PPP

PPP for Solaris 2.6 systems is an asynchronous implementation of the standard data link-level, point-to-point protocol (PPP) included in the internet protocol suite. PPP enables a network administrator to create a communications link using modems and telephone lines. See *TCP/IP and Data Communications Administration Guide* for detailed information about expanding your network with PPP.

UUCP

The Solaris 2.6 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 2.6 due to *job grades*, a mechanism to help sort and prioritize the work load.

Table 12-1 describes the new files and commands offered with Solaris 2.6 UUCP that were not part of the SunOS release 4.x implementation. Table 12-2 describes the log files added to Solaris 2.6 UUCP.

TABLE 12-1 New SunOS release 5.6 UUCP Files and Commands

Command or File	Description
D. 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.
P. data files	
<code>/etc/uucp/Grades</code>	Maps text grade names to system names.
<code>/etc/uucp/Limits</code>	Specifies the number of concurrent UUCP sessions that can occur. Replaces <code>Maxuuscheds</code> and <code>Maxuuxqts</code> files in previous versions.

TABLE 12-1 New SunOS release 5.6 UUCP Files and Commands *(continued)*

Command or File	Description
<code>/etc/uucp/Config</code>	Contains information to override UUCP parameters that can be tuned. Currently, the only parameter of this type is <code>Protocol</code> , so system administrators normally will not have to modify this file.
<code>uuglist</code>	Prints the list of service grades available on the system to use with the <code>-g</code> option of <code>uucp(1C)</code> and <code>uux(1C)</code> .

Solaris 2.6 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 2.6 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 `.P` 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 5.6 UUCP, no comparison can take place and transmissions restarts from the beginning.

User Job Grades

Job grading enables administrators to 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 the SunOS release 4.x software, 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 2.6 operating environment. Solaris 2.6 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 it should normally not be altered by system administrators.

Log Files

Solaris 2.6 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 12-2 Additional SunOS release 5.6 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

TABLE 12-2 Additional SunOS release 5.6 UUCP Log Files *(continued)*

File Name	Function
<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 5.6 UUCP, see *TCP/IP and Data Communications Administration Guide* for complete information.

Using Name Services

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

- “Name Service Switch” on page 123
- “NIS+ ” on page 124
- “DNS ” on page 124
- “DNS and NIS+ Comparison” on page 124
- “NIS and NIS+ Comparison” on page 125
- “Planning an NIS+ Upgrade” on page 127

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

Note - The system administration documentation set for the Solaris 2.6 operating environment emphasizes a system that is using NIS+.

Name Service Switch

The Solaris 2.6 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 2.6 operating environment, which allows applications to use a UNIX standard naming interface (for example, `getxxbyyy` interfaces). 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 13-1 shows the features and benefits of DNS compared to NIS+.

TABLE 13-1 DNS and NIS+ Features and Benefits Compared

Feature	DNS	NIS+
Security	Unrestricted access to data	All operations can be authenticated Administrator designates access rights for objects and entries

TABLE 13-1 DNS and NIS+ Features and Benefits Compared *(continued)*

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

The main strength of DNS is supporting hierarchical database partitions and replicas containing entries of relatively static information (such as host name and IP address). DNS enables you to access the Internet.

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 13-2 summarizes several major enhancements in NIS+ compared to NIS.

TABLE 13-2 NIS and NIS+ Features Compared

Feature	NIS	NIS+
Name space	Has a flat ; is non-hierarchical organization; centralized flat file database for each independent network domain	Has a hierarchical organization; 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
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
Security	Database not secure	Fine-grained access control to NIS+ directories, table column, and entries
Commands and Functions Prefixes	Prefixed by the letters <i>yp</i> , as in <i>ypmatch(1)</i> and <i>ypcat(1)</i>	Prefixed by the letters <i>nis</i> , as in <i>nismatch(1)</i> and <i>nischown(1)</i>

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

Planning an NIS+ Upgrade

NIS+ supports the following combinations of systems:

- SunOS release 5.6 software installed on all servers and clients
- SunOS release 5.6 software installed on one server, but combined with some SunOS release 4.x servers

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

- Upgrade all servers and clients to NIS+
- Upgrade all servers at once to NIS+ and enable its compatibility mode to support SunOS 4.x clients
- Use different domain names so NIS and NIS+ can coexist

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.

Solaris Common Desktop Environment

The Solaris Common Desktop Environment (CDE), compatible among various workstation manufacturers, provides users with a desktop graphical interface on a Sun[™] Workstation[™] running Solaris 2.4 software or later. This window environment helps you organize and manage your work. The desktop provides windows, workspaces, controls, menus, and a Front Panel. When you login to your windows environment the first time, you have a choice of using either OpenWindows or Solaris CDE as your default desktop.

- “What Is the Solaris Common Desktop Environment?” on page 129
- “Overview of the Desktop” on page 130
- “Moving From the OpenWindows Environment to CDE” on page 133

What Is the Solaris Common Desktop Environment?

In March of 1993, Sun, Hewlett-Packard, IBM and Novell announced an agreement to develop a graphical user interface that would bring a consistent look and feel to major UNIX system-based workstations and desktop computers. From the start, the CDE development effort was guided by one goal: to make UNIX easier to use for end users and application developers.

The result of this joint development effort is the Common Desktop Environment. CDE is one of two desktops packaged with the Solaris 2.6 environment (the other is the OpenWindows desktop). Over time, CDE will

emerge as the standard desktop for Sun, Hewlett-Packard, IBM, Novell and many others in the UNIX workstation market.

Solaris CDE includes a desktop server, a Session Manager, a Window Manager (based on Hewlett-Packard's Visual User Environment), and numerous desktop utilities.

To learn how to use Solaris CDE, see *Solaris Common Desktop Environment: User's Guide*.

Developers, End Users, and CDE

Because CDE provides a consistent computing environment across major UNIX platforms, end users can easily move between different machines. CDE also aids application development by supplying a single, standard set of programming interfaces for any conforming Sun, HP, IBM, and Novell platform. A single API enables developers to create applications that are consistent in appearance and behavior across CDE-compliant systems.

The CDE development environment is based on the X11R5 server and produces applications with a look and feel based on the Open Software Foundation's Motif 1.2 specification.

Overview of the Desktop

Some of the features of the Solaris CDE desktop include:

- The Front Panel
- Style Manager
- File Manager
- Application Manager

Front Panel

The Front Panel is a special window at the bottom of the display. It provides controls, indicators, and subpanels you use in your everyday work. The Front Panel also provides the workspace switch for selecting a workspace.

Many controls in the Front Panel, such as the File Manager control, start applications when you click them. Some controls, like the Printer control, are also drop zones. You can drag a file icon from File Manager and drop it on the Printer control to be printed.

Arrow buttons over Front Panel controls identify subpanels—click an arrow button to open a subpanel.

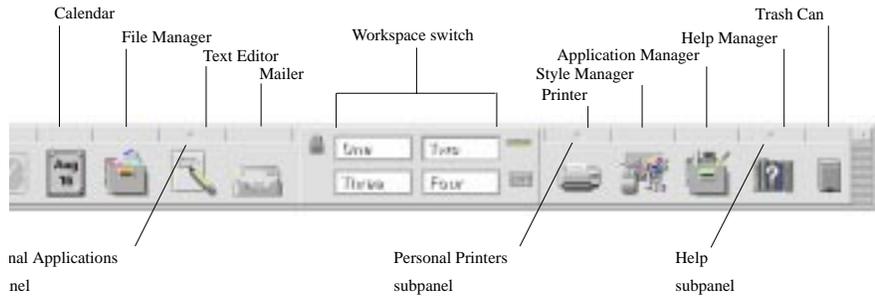


Figure 14-1 Front Panel controls

Style Manager

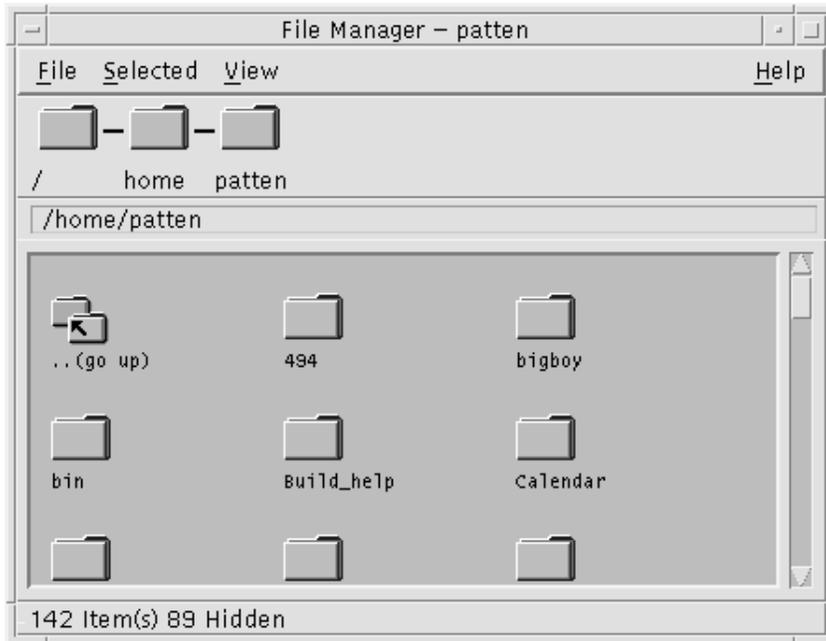


Click this icon:

To use Style Manager to easily customize many elements of the desktop including:

- Colors
- Workspace backdrops
- Font size
- Keyboard, mouse, and window behavior

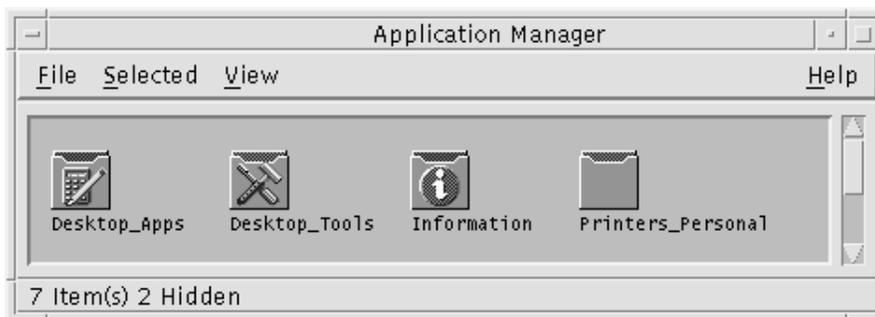
File Manager



Click the File Manager icon:

To display the files, folders, and applications on your system as icons.

Application Manager





Click this icon to open the Application Manager window:

Application Manager provides access to applications you use in your everyday work through *action icons*. You use action icons to start applications. Application Manager stores action icons in special folders called *application groups*.

You can place the action icons you use frequently on the workspace backdrop.

Moving From the OpenWindows Environment to CDE

With the Solaris 2.6 software you can choose to log in to either the OpenWindows desktop or the CDE desktop from your login screen. For more information on logging in, refer to the Login Manager help volume or Chapter 2, “Starting a Desktop Session,” in *Solaris Common Desktop Environment: User’s Guide*.

Desktop Services

Some of the desktop services you are used to using in the OpenWindows environment are located in different places in Solaris CDE. Table 14-1 highlights some of the differences.

TABLE 14-1 Location of Desktop Services

Desktop Service	OpenWindows	CDE
Logout	Workspace menu	Front Panel
LockScreen	Utilities menu	Front Panel
Customize Workspace	Workspace menu	Style Manager
Save Workspace	Utilities menu	Style Manager

TABLE 14-1 Location of Desktop Services (continued)

Desktop Service	OpenWindows	CDE
Refresh	Utilities menu	Front Panel
Properties	Workspace menu	Style Manager
Help	Workspace menu	Front Panel, Application Manager, Workspace menu

Using Windows, Menus, Buttons, and the Mouse in CDE

Windows, menus, buttons and the mouse are used in Solaris CDE slightly differently than in the OpenWindows environment. For complete information on using window, menus, buttons and the mouse, refer to Chapter 1, “Basic Skills,” in *Solaris Common Desktop Environment: User’s Guide*.

Accessing the Workspace Programs Menu

In the OpenWindows environment, the main way to start an application was through the Workspace menu. A Workspace menu still exists in Solaris CDE; however, the main access point for workspace functionality is the Front Panel.

The applications available through the Workspace menu include the items on the Front Panel and also a subset of the applications available to you within Application Manager. Refer to Chapter 6, “Running Applications from the Desktop,” in *Solaris Common Desktop Environment: User’s Guide* for complete information on Application Manager.

Style Manager and Customizing the Workspace

The items available through Style Manager are: Color, Font, Backdrop, Keyboard, Mouse, Audio, Screen, Window, and Startup. This replaces the Workspace Properties window in the OpenWindows environment. For complete information on Style Manager, refer to Chapter 7, “Customizing the

Desktop Environment,” in *Solaris Common Desktop Environment: User’s Guide*.

Running OpenWindows Applications in CDE

A folder in CDE Application Manager, titled OpenWindows, contains your OpenWindow applications.

If you ran OpenWindows applications from the command line, you can run them the same way from the terminal emulator (Terminal application) in Solaris CDE. Refer to Chapter 6, “Running Applications from the Desktop,” in *Solaris Common Desktop Environment: User’s Guide* for complete information on Application Manager.

Application Settings and Properties

In the OpenWindows environment, application-wide settings are set via the Properties dialog box, accessed from the Edit menu. In CDE, application-wide settings are set via the Options areas. Options choices are generally located under the application’s File menu or the separate menu item, Options.

In CDE, Properties (if they exist in an application) are found under the application’s Edit menu and are used to set characteristics of an object, such as its date or name, or display identifying characteristics of an object, such as typefaces. In CDE, format settings are usually found under the Format menu and enable margin and paragraph alignment to be set for a single paragraph, file, or message.

CDE Global options are like the properties you set from the Workspace menu in the OpenWindows environment. You now set these properties from the CDE Style Manager application. See Chapter 7, “Customizing the Desktop Environment,” in *Solaris Common Desktop Environment: User’s Guide*.

Changing Keyboard Defaults

If you did not change your keyboard defaults in the OpenWindows environment they should stay the same within CDE. If you want to change your defaults, use the Style Manager Keyboard dialog box. See Chapter 7, “Customizing the Desktop Environment,” in *Solaris Common Desktop Environment: User’s Guide*. If you need to make changes to your UNIX keyboard bindings, refer to Chapter 10, “Using Text Editor,” in *Solaris Common Desktop Environment: User’s Guide*.

Changing Mouse Defaults

If you did not change your mouse defaults in the OpenWindows environment they should stay the same within CDE. If you want to change your defaults, use the Style Manager Mouse dialog box. Some of the names have been changed for the functions: You still have double-click, acceleration, and threshold. Mouse button order in CDE is called "handedness. See Chapter 1, "Basic Skills," in *Solaris Common Desktop Environment: User's Guide*.

PART II Transition Information for Developers

Changes in the C language and its related tools are among the most obvious differences between the SunOS release 4.x and the Solaris 2.6 operating environments. 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 software. This part of the guide describes these differences, points out similarities between releases, provides information you need to port existing software, or to write new software for the Solaris 2.6 operating environment, and explains the implications for your programming environment.

Compilers, Linkers, and Debuggers

This chapter discusses the changes to compilers, linkers, and debuggers.

- “Compilers ” on page 139
- “Linkers” on page 148
- “Debuggers” on page 156

Compilers

The single most significant change for developers migrating from the SunOS release 4.x to the Solaris 2.6 operating environment is the unbundling of the C compiler. One of the factors that allowed the compiler to be unbundled is the dynamic kernel. The compiler is no longer required to reconfigure the kernel as devices are now automatically loaded by the kernel as needed.

An ANSI C compiler is available with unbundled Sun WorkShop[™]. This compiler produces executables in executable and linking format (ELF), the native object format of Solaris 2.6 executables. `lint` and the `lint` libraries are also unbundled.

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

Compiler Option Differences

Table 15-1 compares the SunOS release 4.x Sun C and ANSI C compilers to the Solaris 2.6 Sun WorkShop™ C++ 4.2. It is provided for your information, and is not intended to describe the range of operations available in default Solaris 2.6 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 means the option is not supported by that driver
- A “+” in the Sun WorkShop™ C++ 4.2 Solaris 2.6 column indicates an option required by SVID specifications.

TABLE 15-1 Comparison of C Compiler Options

Option or Flag	Sun C	ANSI C	Sun WorkShop™ C++ 4.2	Description
-A symbol	No	Yes	Yes	cpp predicate assertion
-a	Yes	Yes	-xa	Counts # basic block executions
-align	Yes	Yes	No	Page aligns (1d)
-assertx	Yes	Yes	-z	Specifies link-time assertion
-BX	Yes	Yes	Yes	Specifies binding type (only for libraries in SunOS release 5.6)
-bnzero	Yes	Yes	No	Generates nonzero AR
-C	Yes	Yes	No	cpp comments left in

TABLE 15-1 Comparison of C Compiler Options *(continued)*

-c	Yes	Yes	+Yes	Produces .o file
-cg87	Yes	Yes	No	Sets fp option to -cg87
-cg89	Yes	Yes	Yes	Sets fp option to -cg89
-dryrun	Yes	Yes	Yes	Shows commands constructed by driver
-Dx	Yes	Yes	+Yes	Defines cpp symbol x
-d	Yes	Yes	No	Forces definition of common (ld)
-dalign	Yes	Yes	Yes	Assumes doubles are double-word aligned
-dl	Yes	Yes	No	Generates long data segment refs
-d[y n]	-BX	-BX	Yes	Dynamic linking [yes no]
-E	Yes	Yes	+Yes	Runs source through cpp
-e	No	No	No	Entry point for ld
-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

TABLE 15-1 Comparison of C Compiler Options *(continued)*

<code>-fstore</code>	Yes	No	Yes	Forces writes on store
<code>-G</code>	No	No	Yes	Creates shared library, not available with the <code>-dn</code> option
<code>-g</code>	Yes	Yes	+Yes	Generates info for dbx
<code>-go</code>	Yes	No	No	Generates info for adb
<code>-H</code>	Yes	Yes	Yes	Prints paths of included files
<code>-h name</code>	No	No	Yes	Uses <code>name</code> as internal identifier; <code>soname</code> passed to linker
<code>-help</code>	Yes	Yes	<code>-flags</code>	Lists options
<code>-Ix</code>	Yes	Yes	+Yes	Adds <code>x</code> to include path
<code>-J</code>	sun3	No	No	Generates long offset for <code>switch case</code>
<code>-KPIC</code>	<code>-PIC</code>	<code>-PIC</code>	Yes	Position independent code
<code>-Kpic</code>	<code>-pic</code>	<code>-pic</code>	Yes	PIC with short offsets
<code>-Kminabi</code>	No	No	No	ABI compliant code
<code>-libmil</code>	Yes	Yes	<code>-xlibmil</code>	Passes <code>libm.il</code> as part of <code>-fast</code>
<code>-lx</code>	Yes	Yes	Yes	Reads object library (for <code>ld</code>)
<code>-Lx</code>	Yes	Yes	Yes	Adds <code>x</code> to <code>ld</code> library path
<code>-M</code>	Yes	Yes	No	Collects dependencies
<code>-M mapfile</code>	Yes	No	No	Passes <code>mapfile</code> to the linker

TABLE 15-1 Comparison of C Compiler Options *(continued)*

<code>-misalign</code>	Yes	Yes	Yes	Handles misaligned Sun-4 data
<code>-N</code>	Yes	No	No	Does not make shared
<code>-n</code>	Yes	No	No	Makes shared
<code>-native</code>	Yes	Yes	Yes	Uses appropriate <code>-cg</code> option
<code>-noc2</code>	Yes	Yes	No	Doesn't do peephole optimization
<code>-nolibmil</code>	Yes	Yes	<code>-xnolibmil</code>	Doesn't pass <code>libm.il</code> with <code>-fast</code>
<code>-o file</code>	Yes	Yes	+Yes	Sets name of output file
<code>-O[1,2,3,4]</code>	Yes	Yes	<code>-xO[1,2,3,4]</code>	Generates optimized code
<code>-O</code>	Yes	Yes	+Yes	Generates optimized code
<code>-P</code>	Yes	Yes	+Yes	Runs source through <code>cpp</code> , output to <code>.i</code>
<code>-PIC</code>	Yes	No	<code>-KPIC</code>	Generates pic code with long offset
<code>-p</code>	Yes	Yes	+Yes	Collects data for <code>prof</code>
<code>-pg</code>	Yes	Yes	<code>-xpg</code>	Collects data for <code>gprof</code>
<code>-pic</code>	Yes	Yes	<code>-Kpic</code>	pic code with short offset
<code>-pipe</code>	Yes	No	No	Uses pipes instead of temp files
<code>-purecross</code>	Yes	No	No	Doesn't have slash in <code>VROOT</code>
<code>-Qdir x</code>	Yes	Yes	<code>-y*</code>	Looks for compiler passes in <code>x</code>

TABLE 15-1 Comparison of C Compiler Options *(continued)*

<code>-Qpath x</code>	Yes	Yes	<code>-Y*</code>	Same as <code>-Qdir</code>
<code>-Qn</code>	No	No	Yes	Doesn't add version stamp info
<code>-Qy</code>	No	No	Yes	Adds version stamp info
<code>-qdir x</code>	Yes	Yes	No	Looks for compiler passes in <code>x</code>
<code>-ql</code>	No	No	No	Collects data for <code>lprof</code>
<code>-qp</code>	<code>-p</code>	<code>-p</code>	<code>-p</code>	Collects data for <code>prof</code>
<code>-qpath x</code>	Yes	Yes	No	Same as <code>-Qdir</code>
<code>-Qoption cpp x</code>	Yes	Yes	Yes	Passes option <code>x</code> on to program <code>cpp</code>
<code>-Qoption iropt x</code>	Yes	Yes	Yes	Passes option <code>x</code> on to program <code>iropt</code>
<code>-Qoption cg x</code>	Yes	Yes	Yes	Passes option <code>x</code> on to program <code>cg</code>
<code>-Qoption inline x</code>	Yes	Yes	No	Passes option <code>x</code> on to program <code>inline</code>
<code>-Qoption as x</code>	Yes	Yes	No	Passes option <code>x</code> on to program <code>as</code>
<code>-Qoption asS x</code>	Yes	Yes	No	Passes option <code>x</code> on to program <code>asS</code>
<code>-Qoption ld x</code>	Yes	Yes	Yes	Passes option <code>x</code> on to program <code>ld</code>
<code>-qoption cpp x</code>	Yes	Yes	No	Passes option <code>x</code> on to program <code>cpp</code>
<code>-qoption ccom x</code>	Yes	Yes	No	Passes option <code>x</code> on to program <code>ccom</code>

TABLE 15-1 Comparison of C Compiler Options *(continued)*

<code>-qoption lint1 x</code>	Yes	Yes	No	Passes option <code>x</code> on to program <code>lint1</code>
<code>-qoption iropt x</code>	Yes	Yes	Yes	Passes option <code>x</code> on to program <code>iropt</code>
<code>-qoption cg x</code>	Yes	Yes	Yes	Passes option <code>x</code> on to program <code>cg</code>
<code>-qoption inline x</code>	Yes	Yes	No	Passes option <code>x</code> on to program <code>inline</code>
<code>-qoption cat x</code>	Yes	Yes	No	Passes option <code>x</code> on to program <code>cat</code>
<code>-qoption c2 x</code>	Yes	Yes	No	Passes option <code>x</code> on to program <code>c2</code>
<code>-qoption as x</code>	Yes	Yes	No	Passes option <code>x</code> on to program <code>as</code>
<code>-qoption asS x</code>	Yes	Yes	No	Passes option <code>x</code> on to program <code>asS</code>
<code>-qoption ld x</code>	Yes	Yes	Yes	Passes option <code>x</code> on to program <code>ld</code>
<code>-Qproduce .o</code>	Yes	Yes	Yes	Produces type <code>.o</code> file (Object file)
<code>-Qproduce .s</code>	Yes	Yes	Yes	Produces type <code>.s</code> file (Assembler source)
<code>-Qproduce .c</code>	Yes	Yes	No	Produces type <code>.c</code> file (C source)
<code>-Qproduce .i</code>	Yes	Yes	Yes	Produces type <code>.i</code> file (C source after <code>cpp</code>)
<code>-qpproduce .o</code>	Yes	Yes	Yes	Produces type <code>.o</code> file (Object file)
<code>-qpproduce .s</code>	Yes	Yes	Yes	Produces type <code>.s</code> file (Assembler source)
<code>-qpproduce .c</code>	Yes	Yes	No	Produces type <code>.c</code> file (C source)

TABLE 15-1 Comparison of C Compiler Options *(continued)*

<code>-qproduce .i</code>	Yes	Yes	Yes	Produces type <code>.i</code> file (C source after <code>cpp</code>)
<code>-r</code>	Yes	Yes	No	Makes relocatable; pass to linker
<code>-R</code>	Yes	Yes	No	Merges data into text segment
<code>-R</code>	No	No	Yes	Specifies search directories for the run-time linker
<code>-S</code>	Yes	Yes	+Yes	Produces <code>.s</code> file only
<code>-s</code>	Yes	Yes	Yes	Strips (4.1); pass to linker
<code>-sb</code>	Yes	Yes	<code>-xsb</code>	Collects information for code browser
<code>-strconst</code>	No	Yes	No	Places string literals in read-only text segment
<code>-sun2</code>	Yes	No	No	Generates code for a Sun-2™ system
<code>-sun3x</code>	Yes	No	No	Generates code for a Sun-3x™ system
<code>-sun386</code>	Yes	No	No	Generates code for a Sun386i™
<code>-sun3</code>	Yes	No	No	Generates code for a Sun-3 system
<code>-sun4c</code>	Yes	No	No	Generates code for a Sun4c™ system
<code>-sun4</code>	Yes	No	No	Generates code for a Sun-4™ system
<code>-target</code>	Yes	No	<code>-xtarget</code>	Sets target architecture to <code>x</code>
<code>-temp=dir</code>	Yes	Yes	Yes	Set directory for temps to <code>dir</code>
<code>-time</code>	Yes	Yes	Yes	Reports the execution times

TABLE 15-1 Comparison of C Compiler Options *(continued)*

-u	Yes	Yes	No	Enters symbol arg as undef (ld)
-Ux	Yes	Yes	+Yes	Undefines <code>cpp</code> symbol x
-v	Yes	Yes	Yes	Verbose mode
-v	No	No	Yes	Strict semantic checking
-V	Yes	Yes	+Yes	Reports versions of programs
-W	No	No	No	Arguments to other components
-w	Yes	Yes	Yes	Does not print warnings
-X[t,a,c,s]	No	Yes	No	Compatibility options
-Y	No	No	No	Changes path name to components
-yx	Yes	No	No	Traces symbol
-z	-assert	-assert	No	Turns on asserts in linker
-#	-v	-v	No	Verbose mode
-EOF	No	No	No	File argument

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

TABLE 15-2 File-name Extensions Used by the C Compiler

Suffix	File Type
.a	Object library
.il	In-line expansion file
.o	Object file
.so	Shared object
.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 2.6 `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 2.6 Replacement	Notes
-align <i>datum</i>	-M <i>mapfile</i>	Uses <i>mapfile</i> and distinct sections
-assert definitions	Default	
-assert nodefinitions	-znodefs	Issues a fatal error instead of a warning
-assert nosymbolic	-zdefs	Issues a fatal error instead of a warning
-assert pure-text	-ztext	Issues a fatal error instead of a warning
-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 151.
-Bnosymbolic	-zdefs	
-Bstatic	-dn & -Bstatic	The -dn option must be specified to completely eliminate the dynamic linker. Use -Bstatic in dynamic mode to include archive libraries. (Used as a toggle. See “Building Executables” on page 151.)
-Bsymbolic	-Bsymbolic	Also gets -assert nosymbolic with this option
-d -dc -dp	Default	Use -b option in SVR4 to turn off

TABLE 15-3 Comparison of ld Options (continued)

-D <i>hex</i>	-M <i>mapfile</i>	<i>mapfile</i> contains 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
-Ldir	-Ldir	<i>dir</i> not recorded in executable; use -R option instead.
-M	-m	
-n	Default	SVR4 executable format compresses disk image as -n
-N	No replacement	
-o <i>name</i>	-o <i>name</i>	
-p	Default	Can override with -M <i>mapfile</i>
-r	-r	
-S	No replacement	
-s	-s	
-t	No replacement	
-T <i>hex</i>	-M <i>mapfile</i>	<i>mapfile</i> contains different mechanisms to accomplish desired effect

TABLE 15-3 Comparison of ld Options (continued)

-Tdata <i>hex</i>	-M <i>mapfile</i>	<i>mapfile</i> contains different mechanisms to accomplish desired effect
-u <i>name</i>	-u <i>name</i>	
-x	No replacement	
-X	No replacement	
-y <i>sym</i>	No replacement	
-z	Default	SVR4 executable format demands pages as -z

Building Shared Libraries

The procedure for building shared libraries in the Solaris 2.6 operating environment requires the `-G` option. In the SunOS release 4.x software, the linker would infer that a shared library was being built by the absence of the `-e` option. As shared libraries may have entry points, this option can no longer be used.

Building Executables

The `-Bdynamic` and `-Bstatic` options are still available, but their behavior is different. These options now refer to library inclusions to the executable rather than binding. Executable binding is set exclusively with the new `-dy` and `-dn` options in the Solaris 2.6 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 2.6 commands that can be used to create similar executables.

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

Specifying Library Search Paths

In the SunOS release 4.x software, 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 `-R` option tells the linker the search paths to be retained for use at run time. See “Search Path Rules” on page 152, 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 runtime linker determine their search paths through a different algorithm from that used by the SunOS release 4.x linker.

The examples below compare the search paths for the dynamic linker and the runtime linker for SunOS release 4.x and the Solaris 2.6 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.

SunOS release 4.x linker search paths:

- Link Editor: `-L, LD_LIBRARY_PATH, /usr/lib, /usr/local/lib`

- Runtime Linker: LD_LIBRARY_PATH, -L, /usr/lib, /usr/local/lib

Solaris 2.6 linker search paths (with LD_LIBRARY_PATH=*dirlist1*):

- Link Editor: -L, *dirlist1*, /usr/ccs/lib, /usr/lib
- Runtime Linker: *dirlist1*, -R, /usr/lib

Solaris 2.6 linker search paths (with LD_LIBRARY_PATH=*dirlist1,dirlist2*):

- Link Editor: *dirlist1*, -L, *dirlist2*, /usr/ccs/lib, /usr/lib
- Runtime Linker: *dirlist1, dirlist2*, -R, /usr/lib

Version Numbering

The SunOS release 4.x software supported both major and minor version numbers on shared libraries. The Solaris 2.6 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 the SunOS release 4.x software.

Table 15-4 shows versions of SunOS release 4.x and Solaris 2.6 shared libraries.

TABLE 15-4 Example Shared Libraries

SunOS release 4.x	Solaris 2.6
libc.so.1.7	libc.so.1
libdl.so.1.0	libdl.so.1

In SunOS release 4.x system software, when the `-l` option was specified, the build environment linker searched for a library with both major and minor numbers. For example, if `-ldl` was specified, the library, `libdl.so.1.0` was linked. In the Solaris 2.6 environment, even though major numbers are still supported, the default behavior of the link editor is to ignore version numbers. Using the previous example, the build environment link editor now searches for `libdl.so` and a symbolic link points to a specific version file.

The recording of a *dependency* in a dynamic executable or shared object is, by default, the file name of the associated shared object as it is referenced by the link editor. To provide a more consistent means of specifying dependencies, shared objects can record within themselves the file name by which they

should be referenced at runtime. This is specified with the `-h` option when linking the library file.

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

Examples

A new utility, `dump(1)`, makes it easier to debug object files or to check the static and dynamic linking, see “Backing Up and Restoring Files” on page 96). 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
```

(Continued)

```
[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

The `dbx` and `dbxtool` tools are no longer available with default system software. Enhanced versions of these tools are available as part of Sun WorkShop[™], an unbundled product.

adb and kadb

The `adb` and `kadb` tools are available in the Solaris 2.6 operating environment. They offer the same capabilities as the SunOS release 4.x tools. `kadb` has been enhanced to recognize multiple processors. The processor ID is displayed in the `kadb` prompt. In the following examples, it is 0.

To make kernel debugging under the Solaris 2.6 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 `kadb` macros described below 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. The following example uses the global unsafe driver `mutex`.

```
kadb[0]: unsafe_driver$<mutex
```

```
kadb[0]: moddebug/W 0x80000000
```

moddebug

`moddebug` enables you to watch module loading. See the end of `<sys/modctl.h>` for legal values for `moddebug` for debugging purposes only.

Debugging a Live Kernel

Use the following command to debug a live kernel.

```
# adb -k /dev/ksyms /dev/mem
```

`/dev/ksyms` is a pseudo device that contains the complete name list of the running kernel.

truss Command

`truss` is a new utility, provided to trace system calls performed, signals received, and machine faults incurred. `truss` offers several significant improvements over the SunOS release 4.x `trace(1)` command, including the ability to follow forked processes and to deal with multithreaded processes.

The following example shows a summary of traced calls for the `date` command. With the `-c` option, `truss` does not display the trace line by line. Instead, it counts the system calls, signals, and faults, and displays a summary.

```
example% truss -c date
Fri Sep 18 14:31:30 PDT 1992
syscall      seconds    calls  errors
_exit        .00        1
read         .00        7
write        .00        1
open         .03        12
close        .00        12
time         .00        1
brk          .01        4
lseek        .00        1
```

(Continued)

(Continuation)

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 for complete details on all `truss` options. There are a number of other Solaris 2.6 debugging tools based on `proc(4)` such as `pmap(1)`.

Tools and Resources

This chapter discusses the changes to tools and resources for the development environment.

- “`ioctl()` Requests” on page 159
- “`ptrace()` Request Values” on page 162
- “Libraries” on page 163
- “Using `make`” on page 167
- “Using SCCS” on page 168
- “Determining Application Compatibility” on page 168
- “Packaging Applications” on page 169
- “Toolkits” on page 171
- “Finding SunOS Release 4.x Tools” on page 171

`ioctl()` Requests

All `ioctl`s related to `dkio(7I)`, `filio`, `mtio(7I)`, `sockio(7I)`, `streamio(7I)`, `termio(7I)`, and `termios(7I)` are supported in this release.

A few incompatibilities exist between the SunOS release 4.x `termios` structure and Solaris 2.6 `termios` structure. For example, the Solaris 2.6 `termios` structure does not include a `c_line` field.

The following `ioctl`s requests, defined in `<sys/ttold.h>`, are not implemented in this release.

- `TIOCMODG`
- `OTTYDISC`

- TABLDISC
- KBLDISC
- TIOCMIDS
- TIOCSETX
- NETLDISC
- NTABLDISC
- TIOCGETX
- NTTYDISC
- MOUSELDISC

The following `ttycom ioctl` requests are not in the Solaris 2.6 operating environment.

- TIOCSCTTY
- TIOCNOTTY
- TIOCISPACE
- TIOCPKT
- TIOCGETPGRP
- TIOCISIZE
- TIOCUCNTL
- TIOCOUTQ
- TIOTCNTL
- TIOCCONS

Table 16-1 shows the `ioctls` supported in the Solaris 2.6 operating environment.

TABLE 16-1 `ioctl()` Support

<code>ioctl()</code>	Description
<code>DKIOCGPART</code>	These requests are replaced with <code>DKIOCGAPART</code> and <code>DKIOCSAPART</code> in Solaris 2.6 software.
<code>DKIOCGCONF</code>	This request is replaced with <code>DKIOCINFO</code> in Solaris 2.6 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 IPI drives. This <code>ioctl</code> fails for SCSI devices. Use the <code>USCSI ioctl</code> for SCSI devices.

TABLE 16-1 `ioctl()` Support (continued)

<code>ioctl()</code>	Description
<code>DKIOCGLOG</code>	<code>EINVAL</code> is returned. <code>DKIOCWCHK</code> toggles the write check on the diskette drive.
<code>filio</code>	The following <code>filio</code> <code>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</code> <code>ioctl</code> requests are not defined in the ABI or SVID.
<code>mtio</code>	Not all devices support all <code>mtio</code> <code>ioctl</code> requests in Solaris 2.6. See the <code>mtio(7)</code> man pages.
<code>sockio</code>	The following <code>sockio</code> <code>ioctl</code> requests are implemented in SVR4 and Solaris 2.6 software: <code>SIOCSPGRP</code> , <code>SIOCGPGRP</code> , <code>SIOCATMARK</code> . <code>sockio</code> <code>ioctl</code> requests are not defined in the ABI or SVID.
<code>streamio</code>	All SunOS release 4.x <code>streamio</code> <code>ioctl</code> requests are implemented in Solaris 2.6 software, the ABI, SVID, and SVR4. The <code>I_FDINSERT</code> request requires an argument that points to a <code>strfdinsert</code> structure. The SunOS release 4.x <code>strfdinsert</code> structure includes an <code>fd</code> (<code>int</code>) field, while the ABI, SVID, or SVR4 <code>strfdinsert</code> structure includes a <code>fildev</code> (<code>int</code>) field instead.
<code>audioio</code>	The SunOS release 4.x <code><sun/audioio.h></code> file has been moved to <code><sys/audioio.h></code> for Solaris 2.6 software. Additionally, in Solaris 2.6 software, there are enhancements to the interface. See the <code>audio(7)</code> , <code>audioamd(7)</code> , or <code>dbri(7)</code> man pages for more information.
<code>termio</code> , <code>termios</code>	All SunOS release 4.x <code>termio</code> and <code>termios</code> <code>ioctl</code> requests are implemented in Solaris 2.6 software, the ABI, SVID, and SVR4. There are a few incompatibilities between the SunOS release 4.x <code>termios</code> structure and Solaris 2.6 software, or the ABI, SVID, or SVR4 <code>termios</code> structure. The SunOS release 4.x <code>termios</code> structure includes a <code>c_line</code> field that is not supported by the other releases. The <code>c_cflag</code> (hardware control of the terminal) can have <code>CRTSCTS</code> (enable RTS/CTS flow control) under the SunOS release 4.x software, but this value is not defined in the Solaris 2.6 software, the ABI, SVID, or SVR4. However, the functionality is supported through the <code>termiox(7)</code> interface.

ptrace() Request Values

The `ptrace()` facility is implemented on top of `/proc`. New applications should use `proc(4)` directly.

The `ptrace()` routine in Solaris 2.6 software is present solely to support applications running in BCP mode. It uses integers 1 – 9 as request values, while the SunOS release 4.x routine defines request values as symbolic constants in `<sys/ptrace.h>`. The following SunOS release 4.x *request* symbolic constants are compatible with Solaris 2.6 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` *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 equivalent Solaris 2.6 request 7 requires that *addr* always be equal to 1 and that execution always resumes from where the process stopped. Also, the Solaris 2.6 request 7 cancels all pending signals before the process resumes execution except those specified by data. The SunOS release 4.x `PTRACE_CONT` does not cancel all pending signals.

Table 16–2 shows SunOS release 4.x valid requests that are not supported by the Solaris 2.6 `ptrace()` routine.

TABLE 16-2 ptrace() Requests not Supported by Solaris 2.6 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_SETWREKPT
PTRACE_GETFPAREGS	PTRACE_SETACBKPT
PTRACE_SETFPAREGS	PTRACE_CLRDR7

Libraries

This release is compliant with the System V Interface Definition, Third Edition (SVID 3). Programs written with the SunOS release 4.1 System V libraries are easy to port to this release. Programs using the SunOS release 4.x BSD C library require more effort.

Reorganized Libraries

Several functions and groups of functions were moved into different libraries. This can cause references to these functions to be flagged as undefined when compiling a SunOS release 4.x application in the Solaris 2.6 environment.

After a compile, check the man page of any functions flagged as undefined. The synopsis list both the `-l` linker option and any include files that you need to resolve the symbol.

Shared Libraries

Shared libraries do not currently support minor version numbers.

Files for shared initialized data (`.sa`) are no longer required; no `.sa` files are provided with the Solaris 2.6 software.

Resource Limits

The Solaris 2.6 operating environment handles resource limits differently. In previous releases, static table allocations were used for resources such as file descriptors and active processes. These resources are now dynamically allocated, so they are limited by the physical memory available. Table 16-3 shows the resource limits.

TABLE 16-3 Resource Limits

Configuration	Limitation
<code>RLIMIT_CORE</code>	Maximum size of core file (in bytes) that can be created by a process
<code>RLIMIT_CPU</code>	Maximum amount of CPU time (in seconds) that a process can use
<code>RLIMIT_DATA</code>	Maximum size of a process's heap (in bytes)
<code>RLIMIT_FSIZE</code>	Maximum size of a file (in bytes) that can be created by a process
<code>RLIMIT_NOFILE</code>	One more than the maximum number of file descriptors that can be created by a process

TABLE 16-3 Resource Limits (continued)

Configuration	Limitation
RLIMIT_VMEM	Maximum size (in bytes) to which a process's mapped address space may grow
RLIMIT_STACK	Maximum size (in bytes) of a process's stack

Note - Any shared objects that need the networking libraries *must* be dynamically linked. The networking libraries require `libdl.so.1`. An archive library is not available.

Table 16-4 shows SunOS release 4.x and Solaris 2.6 libraries and their locations.

TABLE 16-4 Comparison of Library Locations

Library Name	SunOS release 4.x Directory	Solaris 2.6 Directory
<code>libbsdmalloc.a</code>	<code>/usr/lib</code>	<code>/usr/lib</code>
<code>libc.a</code>	<code>/usr/lib</code> and <code>/usr/5lib</code>	<code>/usr/lib</code>
<code>libc.so.1.7</code>	<code>/usr/lib</code>	<code>/usr/lib</code>
<code>libc.so.2.7</code>	<code>/usr/5lib</code>	<code>/usr/lib</code>
<code>libc_p.a</code>	<code>/usr/5lib</code>	Not available
<code>libcurses.a</code>	<code>/usr/lib</code> and <code>/usr/5lib</code>	<code>/usr/ucblib</code> and <code>/usr/ccs/lib</code>
<code>libcurses_p.a</code>	<code>/usr/5lib</code>	Not available
<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>

TABLE 16-4 Comparison of Library Locations *(continued)*

Library Name	SunOS release 4.x Directory	Solaris 2.6 Directory
libg.a	/usr/lib	Not available
libkvm.a	/usr/lib	Not available
libkvm.so.0.3	/usr/lib	/usr/lib
libl.a	/usr/lib	/usr/ccs/lib
libln.a	/usr/lib	Not available
liblwp.a	/usr/lib	Not available
libm.a	/usr/lib	/usr/lib and /usr/lib/libp
libmp.a	/usr/lib	/usr/lib
libnbio.a	/usr/lib	Not available
libnsl.a	/usr/lib	/usr/lib
libpixrect.a	/usr/lib	Not available
libpixrect.so.2.14	/usr/lib	Not available
libposix.a	/usr/lib	Not available
libresolv.a	/usr/lib	/usr/lib
librpcsvc.a	/usr/lib	/usr/lib
libsuntool.so.0.54	/usr/lib	Not available
libsunwindow.so.0.55	/usr/lib	Not available
libsvidm.a	/usr/5lib	Not available

TABLE 16-4 Comparison of Library Locations *(continued)*

Library Name	SunOS release 4.x Directory	Solaris 2.6 Directory
libsvidm_p.a	/usr/5lib	Not available
libtermcap.a	/usr/lib and /usr/5lib	/usr/ucblib and / usr/ccs/lib
libterm.lib.a	/usr/lib and /usr/5lib	/usr/ccs/lib
libxgl.so.1.1	/usr/lib	/opt/SUNWits/ Graphics-sw/xgl/lib
libxpg.a	/usr/xpg2lib	Not available
liby.a	/usr/lib and /usr/5lib	/usr/ccs/lib

Using make

There are two make utilities available in the Solaris 2.6 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, as shown in the following examples.

- In a SunOS release 4.x Makefile – `install`:

```
install -o bin -g bin -m 444 target.c /usr/bin/target
```

- In a SunOS release 5.6 Makefile – `install`:

```
install -u bin -g bin -m 444 target.c /usr/bin/target
```

The version of `install(1B)` in `/usr/ueb` is compatible with the SunOS release 4.x version.

Check the compatibility tables in Appendix A, for information about individual interfaces.

Using SCCS

The Solaris 2.6 operating environment source code control system (SCCS) is slightly different 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 2.6 systems.

In the SunOS release 4.x software, 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 2.6 operating environment.

One difference between SunOS release 4.x and Solaris 2.6 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 2.6 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.

The Binary Compatibility Package provides compatibility for dynamically linked and statically linked applications, as well as hybrids that are partially static and partially dynamically linked.

The Binary Compatibility Package works with well-behaved user applications. Well-behaved applications do not:

- Trap directly to the kernel
- Write directly to any system files

- Use `/dev/kmem`, `/dev/mem`, or `libkvm`
- Use unpublished SunOS interfaces
- Rely on customer-supplied drivers

Applications that are not well-behaved can produce unpredictable results.

Information on using the Binary Compatibility Package is available in *Binary Compatibility Guide*.

Packaging Applications

The Solaris 2.6 operating environment is bundled in units called *packages*. These packages contain all 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`—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 2.6 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 16-5 lists commands that are useful for creating packages.

TABLE 16-5 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 16-6 lists commands that are useful for adding and removing packages.

TABLE 16-6 Commands for Adding and Removing Packages

<code>pkgadd</code>	Adds software package to the system
<code>pkgask</code>	Stores answers to a request script
<code>pkgrm</code>	Removes a package from the system
<code>pkgchk</code>	Checks accuracy of installation

Table 16-7 lists commands that provide information about packages.

TABLE 16-7 Commands for Providing Information About Packages

<code>pkginfo</code>	Displays software package information about installed packages
<code>pkgparam</code>	Displays package parameter values

Toolkits

This section discusses OPEN LOOK Intrinsic Toolkit (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 SunOS release 4.x programming tools are still available and still provide the same capabilities, but many are in new locations. All bundled programming tools are now in the `/usr/ccs/bin` library except `cpp`, which is now in the `/usr/ccs/lib` library. Table 16-8 shows the programming tools and their SunOS release 4.x locations.

TABLE 16-8 Bundled Programming Tools

SunOS release 4.x Command	SunOS release 4.x Location
admin	/usr/sccs
ar	/usr/bin
as	/usr/bin
cdc	/usr/sccs
comb	/usr/sccs
cpp	/usr/lib/cpp
delta	/usr/sccs
error	/usr/ucb
get	/usr/sccs
help	/usr/sccs
ld	/usr/bin
lex	/usr/bin
lorder	/usr/bin
m4	/usr/bin
make	/usr/bin
nm	/usr/bin
prof	/usr/bin
prs	/usr/sccs
prt	/usr/sccs

TABLE 16-8 Bundled Programming Tools *(continued)*

SunOS release 4.x Command	SunOS release 4.x Location
ranlib	/usr/bin
rmdel	/usr/sccs
sact	/usr/sccs
sccs	/usr/ucb
sccsdiff	/usr/sccs
size	/usr/bin
strip	/usr/bin
symorder	/usr/ucb
tsort	/usr/bin
unget	/usr/sccs
unifdef	/usr/ucb
val	/usr/sccs
vc	/usr/old
what	/usr/sccs
yacc	/usr/bin
yaccpar	/usr/lib

Table 16-9 lists the new Solaris 2.6 programming tools and their descriptions.

TABLE 16-9 New Programming Tools

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

Table 16-10 lists the SunOS release 4.x commands that are now unbundled.

TABLE 16-10 Unbundled Programming Tools

Unbundled Command	Description
<code>cb</code>	Simple C program beautifier
<code>cc</code>	C compiler
<code>cflow</code>	Generates a flow graph for a C program
<code>cscope</code>	Interactively examines a C program
<code>ctrace</code>	Generates a C program execution trace
<code>cxref</code>	Generates a C program cross-reference
<code>dbx</code>	Source-level debugger
<code>dbxtool</code>	Window-based source-level debugger
<code>gprof</code>	Displays call-graph profile data

TABLE 16-10 Unbundled Programming Tools *(continued)*

Unbundled Command	Description
<code>indent</code>	Indents and formats C program source files
<code>inline</code>	In-line procedure call expander
<code>lint</code>	C program verifier
<code>objdump</code>	Dumps selected parts of a COFF object file
<code>tcov</code>	Constructs test coverage analysis and statement-by-statement profile

Networking and Internationalization

This chapter discusses Solaris 2.6 networking features as they relate to the programming environment, and it discusses issues concerning the improved internationalization features.

- “Networking” on page 177
- “Internationalization” on page 178

Networking

The Solaris 2.6 operating environment includes the following networking features:

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

See *NIS+ Transition Guide* and *NFS Administration Guide* for more information on using these services.

NIS, NIS+

The Solaris 2.6 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 heterogeneous distributed systems. See “NIS+ ” on page 124 for the nature of NIS+ support available in the Solaris 2.6 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 simplifies 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.

Network Interface Tap

The Network Interface Tap (NIT) provided in the SunOS 4.x release is no longer required. Now Ethernet drivers are real STREAMS drivers that can be opened and communicated with directly.

See `pfmod(7M)`, `bufmod(7M)`, and `dlpi(7P)`

The Solaris 2.6 Ethernet drivers and other data link drivers support the connectionless Data Link Provider Interface (DLPI) Version 2 specification.

Sockets

Sockets are supported in the Solaris 2.6 operating environment. Unlike the SunOS release 4.x software, sockets are no longer implemented completely in the kernel. They are now in a library, `libsocket`, implemented on STREAMS.

Internationalization

Most of the changes in the Solaris 2.6 operating environment improve on previous internationalization features. For complete information on internationalization support, see *Developer's Guide to Internationalization*.

Application developers concerned with internationalizing their programs should follow these guidelines:

- Call `setlocale(3C)` to set up the `LANG` environment variable
- Use standard code sets and follow 8-bit boundaries
- Use `strftime(3C)` to print the date and time
- Replace `strcmp(3)` with `strcoll(3C)` for user-visible collation
- Call `gettext(3C)` or `catget(3C)` to retrieve translated strings from locale-specific message catalogs

Character Support

The Solaris 2.6 operating environment supports extended UNIX code (EUC), VTF8, PCK, and B165. This allows multibyte and multiple code sets on one system.

The SunOS release 4.x software supported single-byte representation of non-ASCII characters. The Solaris 2.6 operating environment supports multibyte representation. This support is needed for Asian language character sets, which contain thousands of characters.

The multibyte functions are included in `libc` and provides the following features:

- Multibyte-to-wide character conversions
- Wide character standard I/O
- Wide character classification
- Wide character formatting

The Solaris 2.6 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 2.6 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.

Locale Database

The SunOS release 5.6 locale database (`/usr/lib/locale/locale`) is completely different than the locale database of SunOS 5.x. This is transparent to the user, however.

Commands

Most of the system commands in the Solaris 2.6 operating environment have been messaged. Many of these commands can pass through multibyte character representations. The increased number of messaged commands makes localization efforts easier.

The `installtxt(1)` command has been replaced with `msgfmt(1)`. Use the new `xgettext(1)` command to extract messages.

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.

`chrtbl(8)` and `catdef(8)` are replaced by `localedef(1)`.

Libraries

The `/usr/xpg2lib/libxpg2.a` archive library is no longer available. These routines have been included in `libc`.

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

TABLE 17-1 xpg2lib Library Routine Locations

Routine	Solaris 2.6 Location
<code>bindtextdomain</code>	<code>/usr/lib/libc</code>
<code>chroot</code>	<code>/usr/lib/libc</code>
<code>catgets</code>	<code>/usr/lib/libc</code>
<code>dgettext</code>	<code>/usr/lib/libc</code>
<code>getcwd</code>	<code>/usr/lib/libc</code>

TABLE 17-1 xpg2lib Library Routine Locations *(continued)*

Routine	Solaris 2.6 Location
getut	/usr/lib/libc
l3tol	Not supported.
logname	/usr/lib/libc
malloc	/usr/lib/libc
swab	/usr/lib/libc
langinfo	/usr/lib/libc
gettext	/usr/lib/libc
sbrk	/usr/lib/libc
textdomain	/usr/lib/libc

Programs that use these routines no longer need to pass `-lxpg2` to the C compiler although some may need to include `libintl.h`. (See Table 17-1 for these routines.)

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 2.6 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.

System and Device Configuration

The operating system kernel and its interfaces have changed significantly. Binary compatibility is not provided for SunOS release 4.x device drivers. This chapter discusses changes in the Solaris 2.6 operating environment that affect kernel and system developers.

- “System Configuration” on page 183
- “Reconfiguration Boot” on page 186
- “Device Naming From a Developer’s Perspective” on page 187

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. They have more limited usage and are

no longer sufficient to correctly install a loadable driver onto the system. `modunload` now includes the capability to unload all unloadable (and not busy) modules. Use `modunload` as follows.

```
# modunload -i 0
```

Kernel Layout

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

The directory search path for modules can be set by the `moddir` variable in the `/etc/system` file (see the `system(4)` man page). Typically, `/platform/'uname -i'/kernel/unix` is the first portion of the kernel to be loaded (see the `kernel(1M)` man page).

config Command

In the SunOS release 4.x software, the `config` command was used to generate system configuration files that enabled `/vmunix` to be relinked from object files. The need for this command has been removed by the following Solaris 2.6 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 the SunOS release 4.x software, `MAXUSERS` was set using `config(8)`. In the Solaris 2.6 operating environment, it is set in the `/etc/system` file with the following line:

```
set maxusers = number
```

Commands that affect loadable modules are of the form:

```
set module:variable=value
```

Changes made to the `/etc/system` file only take effect when you reboot your system (see the `system(4)` man pages).

boot Command

In this release, the following boot programs are available:

- `ufsboot` – To boot from a disk or a CD
- `inetboot` – To boot from across the network

When booting from a disk, the PROM assumes that the primary boot block resides in blocks 1 – 15 of the local disk. Use `installboot(1M)` to create the boot block:

```
# installboot /usr/platform/'uname -i'/lib/fs/ufs/bootblk \  
/dev/rdisk/c0t3d0s0
```

The system firmware loads the primary bootstrap (the boot block) program into memory and runs it. The boot block is a UFS file system reader. It loads the secondary boot program (`/platform/'uname -i'/ufsboot`) into memory.

`ufsboot` loads `kernel/unix`, then `/kernel/unix` uses `ufsboot` to load modules from the `kernel` directory hierarchy until it is able to mount the root file system.

During these operations, the boot block and `ufsboot` use the drivers provided by the firmware; neither `ufsboot` nor the boot block contains any driver code. The `ufsboot` code does not have to change to incorporate a new SBus card with a new disk type since `ufsboot` uses the SBus card PROM driver.

When booting over the network, the boot program performs as it did for a diskless boot in the SunOS release 4.x software. However, the boot program is now called `inetboot` and the client `vfstab` file entries are different. See *System Administration Guide* for information on diskless booting.

Summary of Boot Differences

Table 18–1 summarizes the differences in the boot sequence between the SunOS release 4.x and the Solaris 2.6 operating environment.

TABLE 18-1 Summary of Boot Differences

SunOS release 4.x	Solaris 2.6	Description
boot block	bootblk	Loads ufsboot from disk
boot program	ufsboot	Loads unix from disk
vmunix	unix	Bootable kernel image
boot.sun4c.sunos.4.1.1	inetboot	Mounts and copies unix from network
rc.boot, rc.single	/etc/rcS	Mounts /usr and checks file systems
rc.local	/etc/rc2, /etc/rc3, /etc/rc2.d, /etc/rc3.d	System configuration scripts
config	modload, /etc/system, add_drv, rem_drv	Customizes system kernel; loads, adds, and removes modules as needed
PROM monitor, single user, multiuser	Run states 0 - 6, and S	System run levels

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:

```
ok> boot -r
```

If another device of an existing type (with the driver already installed) is added, and you forget to do a reconfiguration boot, you can use the following commands to tell the system to recognize the new device.

```
# touch /reconfigure
# _INIT_RECONFIG=YES /etc/init.d/drvconfig
# _INIT_RECONFIG=YES /etc/init.d/devlinks
```

Device Naming From a Developer's Perspective

This section expands on the discussion in “Device Naming Conventions” on page 57, 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 “Reconfiguration Boot” on page 186). `drvconfig` configures `/devices` with information about devices (with drivers) that are connected and ready at boot time.

Entries are exported by device drivers calling `ddi_create_minor_node(9F)` when they have determined that a device exists.

Use the `add_drv(1M)` command to add a device to the system. If the driver was successfully added, `add_drv` will also run `drvconfig`.

/dev

In this release, `/dev` is managed by utility programs that create symbolic links to the real entries in `/devices`. The programs are:

- `disks(1M)`
- `tapes(1M)`
- `ports(1M)`
- `devlinks(1M)`

You can run a script to create the appropriate links from `/dev` to `/devices`. The `/dev` names have the advantage of being simpler and

more familiar, while the `/devices` names are unique names for the hardware.

Device Driver Naming

Each device in the system is driven by a device driver. Device drivers manage many instances of a device. Devices are named in several ways:

- Physical names
- Logical names
- Instance names

Physical Names

Physical names are stored in `/devices`. They describe the hardware, and vary with the platform and configuration. For example:

```
/devices/vme/xdc@6d,ee80/xd@0,0:g
```

Physical names can be used to identify which piece of hardware is in use. For example, `xdc@6d,ee80` refers to the disk controller at address `0xee80` in VME A16, D32 space. See the `vme(4)` and `driver.conf(4)` man pages.

Logical Names

Logical names are stored in `/dev`. They attempt to abstract most of the nature of physical device names that are specific to the platform. Logical names might be appropriate for an `xd` device, such as:

```
/dev/dsk/c2d0s6 (controller 2, slave 0, slice 6 (4.x partition "g"))
```

or an `sd` device, such as:

```
/dev/dsk/c0t3d0s0 (controller 0, target 3, lun 0, slice 0 (4.x partition "a"))
```

The logical name conveys nothing about the type of controller. 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 the SunOS release 4.x software.

Disk names also use the SVR4 convention of `/dev/dsk/*` for block disk devices and `/dev/rdisk/*` for raw disks. For more information, see *System Administration Guide*.

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 example.

```
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.

Device Drivers and STREAMS

This chapter discusses device driver issues such as changes to device driver interfaces, the `devinfo` command, porting considerations, STREAMS, and Solaris 2.6 driver architecture.

- “Device Drivers and STREAMS Device Drivers” on page 191
- “Device Driver Commands” on page 198

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

- *STREAMS Programming Guide*
- *System Interface Guide*
- *System Administration Guide*

Device Drivers and STREAMS

Device Drivers

Some of the many changes to device drivers in the Solaris 2.6 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.

Unlike previous releases of SunOS systems (SunOS release 4.1.3 software and earlier), the device driver interfaces in the Solaris 2.6 operating environment are formalized and are referred to as the *Solaris 2.6 SPARC DDI/DKI*. The Solaris 2.6 SPARC DDI/DKI provides binary compatibility of device drivers across all supported platforms and for all future releases of the Solaris 2.6 operating environment on those platforms.

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 they are 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.

With this level of binary compatibility, third-party hardware developers can now “shrink-wrap” their DDI-compliant device drivers with their driver hardware. Installing a new driver package can now be entirely automated. The self-configuring kernel removes the necessity for recompiling the kernel to add or remove a driver. Thus, DDI-compliant device driver for Solaris 2.6 environments can be treated like any other consumer software product.

In the Solaris 2.6 DDI/DKI the DDI-only interfaces are generic to all systems that support the Solaris 2.6 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 2.6 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 2.6 DDI/DKI compliant*. A Solaris 2.6 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 sections of man Pages(1M): System Administration Commands listed below. 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

A Device Driver Developers Kit (DDK) is available separately.

devinfo Command

The Solaris 2.6 `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)

```

(Continued)

```

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 2.6 drivers will look more like SBus drivers than other types. All drivers are loadable, and no kernel configuration is required.

Under the SunOS release 4.x software, 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 2.6 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. 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 pre-emption, 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.

1. Strictly speaking, the splN/splr pair do work; however, although they will block interrupts, the effect is useless in protecting data structures in a multiprocessor environment.

A large percentage of the driver code for simple device drivers, which consist primarily of calls to kernel interface routines, will change, but in straightforward ways. For complex device drivers (such as a SCSI driver) which contain large amounts of device-specific handling code, 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 a driver in the Solaris 2.6 operating environment, refamiliarize yourself with how the driver works. Determine what the SunOS release 4.x driver *did* (not the specific implementation, but general behavior). What interfaces did it export? What `ioctl`(s) did it provide? How did the hardware work and what peculiarities of the hardware did the driver support? Did the driver support multiple `open`() calls?

The following changes affect drivers and should be considered:

- The entry points to drivers are very different
- ANSI C requirements:
 - `volatile` keyword
 - `const` keyword
 - Function prototype declarations
- Relocated or renamed header files (most, if not all, system header files are now in `/usr/include/sys`)
- Most structures have become opaque or are no longer needed. For example:
 - `struct user`
 - `struct proc`
 - `struct dev_info`

STREAMS

Some areas of change for STREAMS modules are transparent I/O controls, automatic pushing of modules on a stream, and new message types.

Transparent `ioctl`(s)

In the SunOS release 4.x software, 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 `strioc_t` structure and then use:

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

There was no easy way to determine whether a driver was STREAMS-based. Now, unrecognized `ioctls` to the stream head are passed on to the driver, eliminating the need to know whether a driver was STREAMS-based.

Message types added in the Solaris 2.6 software support transparent `ioctls`. There are now “copy in” and “copy out” messages to inform the STREAM head to transfer user data to and from the kernel.

For more information on writing STREAMS drivers, see the *STREAMS Programming Guide*.

autopush Command

The SunOS release 4.x `streamtab` structure enabled a driver to specify that certain STREAMS modules be pushed when the device was `open()`.

In the Solaris 2.6 operating environment, the system administrator and the `autopush(IM)` command specify when a STREAMS module is pushed. If required, `autopush` can be run at driver installation.

See *STREAMS Programming Guide* for more information about pushing STREAMS modules.

Solaris 2.x Driver Architecture

To achieve binary compatibility across all currently supported hardware platforms, the DDI interfaces were carefully designed around architectural abstractions. The underlying abstraction, the `device` tree, is an extension of the `devinfo` tree in the original SPARCstation™ design. Each node in the device tree is described by a device information structure or “`dev_info` node.” The bottom-most nodes in the tree are termed *leaf nodes*. Most devices (such as disks and tape drives, framebuffers, I/O cards, and network interfaces) are examples of leaf devices that would be associated with leaf nodes. The associated device drivers are called *leaf drivers*.

The intermediate nodes in the tree are generally associated with buses (for example, SBus, SCSI, VME). These nodes are called *nexus nodes* and the drivers associated with them are called *nexus drivers*. Bus nexi are intended to encapsulate the architectural details associated with a particular element.

Currently, the Solaris 2.6 DDI/DKI supports only the writing of *leaf drivers* and one type of *nexus driver*, the SCSI host bus adapter driver.

The device tree structure creates a formal parent-child relationship between nodes. This parent-child relationship is the key to platform architecture independence.

When a leaf driver requires a service that is platform dependent (for example, a DMA mapping), the system transparently converts the request into a call to its parent to provide the service. The service providers are always nexus drivers; each nexus driver can in turn pass the request to its parent in order to provide the service. This approach enables 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 2.6 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 2.6 command is provided.
- If an interface is listed as the same (S), the Solaris 2.6 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 5.6 column.

Note - The SunOS release 5.6 directory structure is different than the SunOS release 4.x structure; some commands behave the same, but have a different path name. For example, the SunOS release 4.x `/usr/etc/newfs` command now resides in `/usr/sbin/newfs`, but the interface has not changed. This command, and others like it, are considered the same (S) according to this table's guidelines.

Commands that exist in both `/usr/bin` and `/usr/5bin` have two table entries, the first documents the `/usr/bin` command, and the second entry documents the `/usr/5bin` command.

For complete information on all Solaris 2.6 interfaces, see `man Pages(1): User Commands`.

Examples

Table A-1 through Table A-4 show sample table entries and are followed by an interpretation

TABLE A-1 Table Entry Example 1

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
<code>fasthalt(8)</code>	A	The <code>init 0</code> command provides similar capabilities	S

The `fasthalt` command is not available in the Solaris 2.6 base release. This command is available if you install the SunOS/BSD Compatibility package on your system. The `init 0` command replaces `fasthalt`. If you use the compatibility package `fasthalt` command in scripts or applications, they will not work on other SVR4 systems. Compatibility package commands can be found in `/usr/ucb` on systems that have this package installed, and they are documented in section 1B of `man Pages(1): User Commands`; for example, `fasthalt(1B)`.

TABLE A-2 Table Entry Example 2

SunOS Release 4.x Command	SunOS release 5.6 Status	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 5.6 software. A C compiler is available with the SunOS/BSD Compatibility package, but it requires the unbundled C compiler and does not provide the same interface and output as the SunOS release 4.x compiler. See “Compiler Option Differences” on page 140 for details.

TABLE A-3 Table Entry Example 3

SunOS Release 4.x Command	SunOS release 5.6 Status	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 5.6. See the date(1) man page for more information.	N

The SunOS release 4.x software 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 5.6 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 5.6 software. If you are accustomed to using the SunOS release 4.x /usr/bin/date command, you should look at the SunOS release 5.6 date(1) man page before attempting to set the date on your system.

TABLE A-4 Table Entry Example 4

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
rev(1)	N		N

The SunOS 4.2 rev command is not available in the SunOS release 5.6 software or the BSD release. There is no replacement command available.

Commands Reference Table

Table A-5 lists all SunOS release 4.x command interfaces, and shows their status in the Solaris 2.6 environment and in the SunOS/BSD Source Compatibility Package.

TABLE A-5 Commands Reference Table

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
C2conv(8)	N	See your system vendor for information on this product.	N
C2unconv(8)	N	See your system vendor for information on this product.	N
Mail(1)	A	The mailx(1) command provides similar capabilities.	N
ac(8)	A	The System Accounting Resource package (SAR) provides most of the accounting capabilities available in ac.	N
acctcms(8)	S		N
acctcom(8)	S		N
acctcon1(8)	S		N
acctcon2(8)	S		N
acctdisk(8)	S		N
acctdusg(8)	S		N
acctmerg(8)	S		N
accton(8)	S		N
acctprcl(8)	S		N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
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 swmttool(1M) command provides similar capabilities.	N
addbib(1)	S		N
adjacentscreens(1)	A	The OpenWindows environment has two methods for providing multiple displays: (1) Start two servers on a given machine, each server controlling its specific display. (2) Start one server with two displays, using the openwin -dev option.	N
admin(1)	C	The following SunOS release 4.x options are not available in the SunOS release 5.6 system software: -lrelease[,release . . .]Lock indicated release against deltas.	N
adv(8)	N	RFS does not exist. This capability is still accessible via the -f flag.	N
aedplot(1G)	N		S
align_equals(1)	A	The OpenWindows Text menu Indent command provides similar capabilities.	N
analyze(8)	A	Use adb(1) on core files to analyze crashes.	N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
apropos(1)	C	The SunOS release 4.x command used the <code>what is</code> database. In the SunOS release 5.6 software, this database is called <code>windex</code> , and the format is slightly different.	N
ar(1V)	S		N
ar(1V) - SysV	C		N
arch(1)	C	Without options, this command now returns "sun4." Its use is discouraged. Use <code>uname(1)</code> instead. To determine the operating system name and release level, use <code>uname -sr</code> .	S
arp(8C)	S		N
as(1)	C	The following SunOS release 4.x options are not available in the SunOS release 5.6 command: <code>-d2,-h,-j,-J,-k,-O[n]</code> .	N
at(1)	S	The <code>at</code> , <code>atq</code> , and <code>atrm</code> commands in SunOS release 5.6 systems behave slightly differently than they do in SunOS release 4.x systems. Security for non-privileged users is more restricted on SunOS release 5.6 systems. Non-privileged users cannot display the jobs of any other user.	N
atoplot(1G)	N		S

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
atq(1)	C	The <code>at</code> , <code>atq</code> , and <code>atrm</code> commands in SunOS release 5.6 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 5.6 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 5.6 systems.	N
atrm(1)	C	The <code>at</code> , <code>atq</code> , and <code>atrm</code> commands in SunOS release 5.6 systems behave slightly differently than they do in SunOS release 4.x systems. The SunOS release 4.x <code>-'</code> flag has been renamed to <code>-a</code> in the SunOS release 5.6 command. Security for non-privileged users is more restricted on SunOS release 5.6 systems.	N
audit(8)	C	<code>-d</code> or <code>-u</code> 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

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
automount(8)	C	The following SunOS release 4.x option is not available in the SunOS release 5.6 command: -m Suppress initialization of directory-map pairs. The auto.master and auto.home files are renamed auto_master and auto_home. The default home directory path is /export/home/ <i>username</i> .	N
awk(1)	S		N
banner(1V) - SysV	S		N
bar(1)	tar, cpio	The tar(1) command can replace bar for most uses. You can use cpio -iH bar to restore existing SunOS release 4.x bar backups. You can no longer create bar format files.	N
basename(1)	S	The SunOS release 5.6 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 5.6 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 5.6 batch job <i>queuename</i> is not specified. Jobs were always queued on queue b with the SunOS release 4.x command.	N
bc(1)	S		N
bgplot(1G)	N		S
biff(1)	chmod	fiff n: % chmod u+x 'tty' biff y: % chmod u-x 'tty'	S

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
bin-mail(1)	S	Same as the SunOS release 5.6 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)	C	An OpenWindows version of this command is available with the OpenWindows text editor.	N
captainfo(8V) - SysV	S		N
cat(1V) - SysV	S		N
cat(1V)	S	The SunOS release 5.6 cat command requires the -v option with the -t and -e options. The SunOS release 5.6 command displays FORMFEED characters with the -t option, instead of the -v option as with the SunOS release 4.x command.	N
catman(8)	S		N
cb(1)	S		N
cc(1V) - SysV	N		N
cc(1V)	N	The C compiler is only available with the C language unbundled tools.	C
cd(1)	S		N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
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 5.6 command silently ignores the error.	N
cflow(1V) - SysV	N	The cflow command is now available as an unbundled product.	N
cflow(1V)	N	The cflow command is now available as an unbundled product.	N
chargefee(8)	S		
checkeq(1)	S		N
checknr(1)	S		N
chfn(1)	N		N
chgrp(1)	C	The default behavior of symbolic links has changed from SunOS release 4.x to SunOS release 5.6 system software. In SunOS release 4.x system software, chgrp changed ownership of the symbolic itself; in SunOS release 5.6 system software, chgrp follows the link. To change ownership of the symbolic link in SunOS release 5.6 system software, use the -h option.	N
chkey(1)	S		N
chmod(1V) - SysV	C	The SunOS release 5.6 -R option changes the mode of the target when symbolic links are encountered.	N
chmod(1V)	S	The SunOS release 5.6 -R option changes the mode of the target when symbolic links are encountered. The SunOS release 5.6 command supports two additional permissions: 'l' and 't'.	N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
chown(8)	C	The default behavior of symbolic links has changed. SunOS release 4.x <code>chown</code> changed ownership of the symbolic link. SunOS release 5.6 <code>chown</code> follows the link. To change the ownership of the link, use <code>chown -h</code> . The SunOS release 5.6 <code>chown</code> command does not allow changing the group ID of a file.	S
chroot(8)	S		N
chrtbl(8)	A	In SunOS release 5.6 <code>localedef(1)</code> creates locale database.	N
chsh(1)	N		N
ckpacct(8)	S		N
clear(1)	S		N
clear_colormap(1)	N		N
clear_functions(1)	S		N
click(1)	N		N
clock(1)	A	An OpenWindows command is available in <code>/usr/demo/clock</code> . See the <code>clock(1)</code> man page for information.	N
clri(8)	S		N
cmdtool(1)	A	This command is replaced by the OpenWindows Command Tool.	N
cmp(1)	S		N
col(1V) - SysV	S		N
col(1V)	C		N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
colcrt(1)	N		N
colldef(8)	A	In SunOS release 5.6 <code>localedef(1)</code> creates locale database.	N
coloredit(1)	A	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 <code>s.file</code> . The SunOS release 4.x command prints an error, but the SunOS release 5.6 command silently ignores the error.	N
comm(1)	S		N
compress(1)	S		N
config(8)	N		N
cp(1)	C	The <code>-R</code> option is replaced by the <code>-r</code> option in the SunOS release 5.6 command.	N
cpio(1)	S		N
cpp(1)	S		N
crash(8)	C	The default name list used in SunOS release 4.x is <code>/vmunix</code> , but it is <code>/kernel/unix</code> in the SunOS release 5.6 software.	N
cron(8)	S		N
crontab(1)	S		N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
crtplot(1G)	N		S
crypt(1)	S		N
cs(1)	S		N
csplit(1V) - SysV	S		N
ctags(1)	S		N
ctrace(1V) - SysV	N	<p>The following SunOS release 4.x option is not available in the SunOS release 5.6 command:</p> <p>-b 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.</p> <p>The syntax of the <code>-r</code> option differs between SunOS release 4.x and SunOS release 5.6 system software. The 4.1 format is <code>-rf</code>; it is now <code>-r f</code>. <code>ctrace</code> is available as an unbundled product.</p>	N
cu(1C)	S		N
cut(1V) - SysV	S		N
cxref(1V) - SysV	S		N
cxref(1V)	N	<code>cxref</code> is available as an unbundled product.	N
date(1V) - SysV	S		N
date(1V)	C	The format used when setting the date is slightly different in the SunOS release 5.6 software. See the <code>date(1)</code> man page for more information.	N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
dbconfig(8)	S		N
dbx(1)	N	Available with the unbundled SPARCworks product.	N
dbxtool(1)	N	Available with the unbundled SPARCworks product as the command debugger.	N
dc(1)	S		N
dcheck(8)	A	Use the <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
dd(1)	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 5.6 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.	N
defaults_from_input(1)		The function of this command is now handled by the OpenWindows property window.	N
defaults_merge(1)	S		N
defaults_to_indentpro(1)		The function of this command is now handled by the OpenWindows property window.	N
defaults_to_mailrc(1)		The function of this command is now handled by the OpenWindows property window.	N
defaultsedit(1)		The function of this command is now handled by the OpenWindows property window.	N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
delta(1)	C	If a directory is specified as the argument, all files in the directory are processed. In the SunOS release 4.x software, an error is produced if a file in a directory generates an error. Such files are silently ignored by the SunOS release 5.6 command.	N
deroff(1)	S		N
des(1)	S		N
devinfo(8S)	C	The <code>prtconf(1M)</code> command provides similar capabilities.	N
devnm(8)	C	The output format between SunOS release 4.x and SunOS release 5.6 system software is quite different. In SunOS release 4.x system software, the name argument is optional. In the SunOS release 5.6 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 5.6 <code>df</code> command. The SunOS release 5.6 <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 5.6 <code>df -t</code> command prints full listings with totals. You can use <code>df -l</code> to see local file systems.	S

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	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 5.6 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 5.6 command: <code>-da sec</code> Disassemble <code>sec</code> as data, printing the actual address of the data. Use the SunOS release 5.6 <code>-D sec</code> option to do the same thing.	N
diskusg(8)	A	The <code>acctdusg(1M)</code> command provides similar capabilities.	N
dkctl(8)	N		N
dkinfo(8)	A	The <code>prtvtoc(1M)</code> command provides similar capabilities.	N
dmesg(8)	S		N
dname(8)	N	RFS is not available.	N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
dodisk(8)	S		N
domainname(1)	S		N
dorfs(8)	N	RFS is not available.	N
dos2unix(1)	S		N
du(1V) - SysV	S		N
du(1V)	C	The SunOS release 4.x command reports the disk usage in kilobytes while the SunOS release 5.6 du command reports disk usage in 512-byte blocks. The -k option can be used to report usage in kilobytes.	S
dumbplot(1G)	N		S
dump(8)	A	The <code>ufsdump</code> command provides similar capabilities. The following SunOS release 4.x options are not in the SunOS release 5.6 command: -a <i>archive-file</i> The SunOS release 5.6 -a option dumps the archive header of each member of an archive. -D Specify diskette as the dump media. The SunOS release 5.6 -D option dumps debugging information. -v Verify against the file system being dumped. The SunOS release 5.6 -v option dumps information in symbolic, rather than numeric, representation.	N
dumpfs(8)	A	The <code>fstyp -F -ufs -v</code> command provides similar capabilities.	N
dumpkeys(1)	S		N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
e(1)	A	The <code>ex(1)</code> command provides similar capabilities.	S
echo(1V) - SysV	S		N
echo(1V)	C	The SunOS release 4.x <code>-n</code> option suppressed new-line printing. Use a <code>\c</code> in the SunOS release 5.6 software.	S
ed(1)	S		N
edit(1)	S		N
edquota(8)	S		N
EEPROM(8S)	S		N
egrep(1V)	S		N
eject(1)	S		N
enroll(1)	N		N
env(1)	S		N
eqn(1)	S		N
error(1)	S		N
etherd(8C)	A	The <code>snoop(1M)</code> command provides similar capabilities.	N
etherfind(8C)	A	The <code>snoop(1M)</code> command provides similar capabilities.	N
ex(1)	S		N
expand(1)	S		N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
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 5.6 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 5.6 command: -n <i>cpio-device</i> Write the current file on device in <i>cpio -c</i> format.	N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
finger(1)	S		N
fingerd(8)	S		N
fmt(1)	C		N
fmt_mail(1)	N		N
fold(1)	S		N
fontedit(1)	N		N
foption(1)	N		N
format(8S)	S		N
fpa_download(8)	N		N
fparel(8)	N		N
fpaversion(8)	N		N
fpurel(8)	N		N
fpuversion4(8)	A	This information is available from psrinfo -v	N
from(1)	N		S
fsck(8)	C	The SunOS release 4.x fsck command differs significantly from the SunOS release 5.6 command. With the SunOS release 5.6 command, you specify most options after you specify the file system type. fsck -m does a quick file-system check. The -w option is not available. New options include -f, -v, and -o.	N
fsck-cdrom(8)	N		N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
fsirand(8)	S		S
ftp(1C)	S		N
ftpd(8C)	S		N
fumount(8)	S	RFS is no longer available	N
fusage(8)	S	RFS is no longer available	N
fuser(8)	S		N
fwtmp(8)	S		N
gcore(1)	S		N
generic_args(1)	N		N
get(1)	C	The SunOS release 5.6 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 5.6 command ignores them silently.	N
get_alarm(1)	N		N
get_selection(1)	A	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

TABLE A-5 Commands Reference Table *(continued)*

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
getty(8)	S		N
gfxtool(1)	N		N
gigiplot(1G)	N		S
glob(1)	S		N
goto(1)	S		N
gpconfig(8)	N		N
gprof(1G)	S		N
graph(1G)	S		N
grep(1V)	S		N
grep(1V) - SysV	C	The following option has changed: -w Search for the regular expression as a word as if surrounded by \< and \>.	N
groups(1)	S		S
grpck(8V)	S		N
gxtest(8S)	N		N
halt(8)	S		N
hashcheck(1)	S		N
hashmake(1)	S		N
hashstat(1)	S		N
head(1)	S		N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
help(1)	S		N
help_open(1)	S		N
hostid(1)	S		S
hostname(1)	S		S
hostrfs(8)	N	RFS is not available.	N
hp7221plot(1G)	N		S
hplot(1G)	N		S
htable(8)	S		N
i386(1)	S		N
iAPX286(1)	S		N
icheck(8)	A	fsdb() is an alternate command.	N
iconedit(1)	A	This command is replaced by the OpenWindows Icon Edit tool.	N
id(1)			
id(1V) - SysV	S		N
idload(8)	N	RFS is not available.	N
ifconfig(8C)	S		N
imentest(8C)	N		N
implot(1G)	N		Y
in.comsat(8C)	S		N

TABLE A-5 Commands Reference Table *(continued)*

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
<code>in.fingerd(8C)</code>	S		N
<code>in.ftpd(8C)</code>	S		N
<code>in.named(8C)</code>	S		N
<code>in.rexecd(8C)</code>	S		N
<code>in.rlogind(8C)</code>	S		N
<code>in.routed(8C)</code>	S		N
<code>in.rshd(8C)</code>	C	The port range differs between the SunOS release 4.x and SunOS release 5.6 commands. In SunOS release 4.x system software, the range is 512-1023; in SunOS release 5.6 system software, it is 0-1023.	N
<code>in.rwhod(8C)</code>	S		N
<code>in.talkd(8C)</code>	S		N
<code>in.telnetd(8C)</code>	S		N
<code>in.tftpd(8C)</code>	S		N
<code>in.tnamed(8C)</code>	S		N
<code>in.uucpd(8C)</code>	S		N
<code>indent(1)</code>	N	This command is now available as an unbundled product.	N
<code>indentpro_to_defaults(1)</code>	A	The function of this command is now handled by the OpenWindows property sheets.	N
<code>indxbib(1)</code>	S		N
<code>inetd(8C)</code>	S		N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
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 5.6 system software. In the SunOS release 5.6 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 5.6 command is very different from the SunOS release 4.x command. See the <code>init(1M)</code> man page for more information.	N
inline(1)	N	This command is now available as an unbundled product.	N
input_from_defaults(1)	N		N
insert_brackets(1)	A	An OpenWindows command with the same name is available with the OpenWindows Text Editor.	N
install(1)	C	The functions of the <code>-c</code> , <code>-o</code> , and <code>-s</code> options are different between the SunOS release 4.x and SunOS release 5.6 commands.	S
installboot(8S)	C	The path names and syntax have changed.	N
installtxt(8)	A	The <code>msgfmt(1)</code> command provides similar capabilities.	N
intr(8)	N		N
iostat(8)	S	New options: <code>-x</code> Provide disk statistics <code>-c</code> Report the percentage of time the system has spent in user mode, system mode, and idle.	N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
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 the SunOS release 5.6 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 5.6 command.	N
kadb(8S)	S		N
keyenvoy(8C)	N		N
keylogin(1)	S		N
keylogout(1)	S		N
keyserv(8C)	S		N
kgmon(8)	S		N
kill(1)	S		N
labelit(8)	S		N
last(1)	S		N
lastcomm(1)	S		N
lastlogin(8)	S		N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
ld(1)	C	There are many differences between the SunOS release 4.x ld command and the SunOS release 5.6 command. The following SunOS release 4.x options are not available: -align, -A, -B, -D, -M, -n, -t, -T, -Tdata, -x, -X, -y and -z. The -assert option has been replaced by the -z option. The -d, -dc, -dp options are the default in SunOS release 5.6 system software. To turn off these options use -b.	S
ldconfig(8)	N		N
ldd(1)	S		N
leave(1)	N	The cron(1M) and at(1) commands provide similar capabilities.	N
lex(1)	C	The following SunOS release 4.x option is not available in the SunOS release 5.6 command: -f 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 the unbundled SPARCworks product.	S
listen(8)	S		N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
ln(1V)	C	The SunOS release 4.x ln command never removes the target if it already exists. The SunOS release 5.6 ln command removes the target, given the proper permissions. The SunOS release 4.x -f option forces a hard link to a directory.	S
ln(1V) - SysV	C	In SunOS release 4.x /usr/5bin/ln, the -f option forces files to be linked without displaying permissions, asking questions, or reporting errors. The /usr/5bin/ln -F option to force a hard link to a directory is not available in SunOS release 5.6 system software.	N
loadkeys(1)	S		N
lockd(8C)	S		N
lockscreen(1)	A	This command is available as the OpenWindows tool xlock(1). The capabilities of the lockscreen command remains the same in xlock, although the foreground pattern differs.	N
logger(1)	N		S
login(1)	S		N
logname(1)	S		N
look(1)	S		N
lookbib(1)	S		N
lorder(1)	S		N
lp(1)	S		N

TABLE A-5 Commands Reference Table *(continued)*

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
lpc(8)	A	The lpadmin(1M) command provides similar capabilities.	S
lpd(8)	A	The lpadmin(1M) command provides similar capabilities.	S
lpq(1)	A	The lpstat(1) command provides similar capabilities.	S
lpr(1)	A	The lp(1) command provides similar capabilities.	S
lprm(1)	A	The cancel(1) command provides similar capabilities.	S
lpstat(1)	S		N
lptest(1)	N		S
ls(1V) – SysV	C		N
ls(1V)	S		S
lsw(1)	N		N
m4(1V)	C	Some small syntactic incompatibilities over expression evaluation.	N
m4(1V) – SysV	S		N
m68k(1)	S		N
mach(1)	S		S
mail(1) – UCB	mailx		S

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
mail(1)	C	Now in <code>/usr/bin/mail</code> , was in <code>/usr/ucb/mail</code> in the SunOS release 4.x software. This entry refers to the mail command installed under <code>/usr/bin/mail</code> . The SunOS release 4.x mail is compatible with the SunOS release 5.6 command except for the following: -i The <code>-i</code> (ignore interrupts) option is not available. In the SunOS release 4.x command, the postmark line is preceded by a '>'; this is not required by the SunOS release 5.6 command.	N
mailrc_to_defaults(1)	C	The function of this command is now handled by the OpenWindows property window.	N
mailstats(8)	S		N
mailtool(1)	C	This command is available as the OpenWindows Mail Tool.	N
make(1)	S	SVR4 & SVID make is available in <code>/usr/ccs/lib/svr4.mke</code>	N
makedbm(8)	C	The SunOS release 5.6 interface for this command is compatible with the SunOS release 4.x interface. The SunOS release 5.6 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

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
man(1)	C	<p>The organization of the on-line man pages has changed. Refer to intro(1) for a description of all sections. The man command now allows you to specify a default order of directories for man to search. Two new options make it easier to find man pages:</p> <p>-a to display all man pages matching <i>title</i> in the order found; and</p> <p>-l to list all man pages matching <i>title</i>.</p> <p>Also, the -s option replaces the <i>section number</i> argument.</p>	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	<p>The interface differs significantly between the two versions. The SunOS release 5.6 command provides for different file system types.</p>	N
mknod(8)	S		N
mkproto(8)	C		N
mkstr(1)	N		S
modload(8)	C	<p>Modules are usually automatically loaded using modload.</p>	N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
modstat(8)	A	The modinfo(1M) command provides similar capabilities.	N
modunload(8)	C	Modules are usually automatically unloaded.	N
monacct(8)	S		N
more(1)	S		N
mount(8)	C	The interface differs significantly between the two versions. In the SunOS release 5.6 version, most options must be specified after the file system type has been specified (unless the file system is entered in /etc/vfstab).	N
mount_tfs(8)	N		N
mountd(8C)	S		N
mt(1)	S		N
mv(1)	S		N
named(8C)	C	The name daemon is renamed to 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
netstat(8C)	S		N
newaliases(8)	S		N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
newfs(8)	S	Moved from /usr/etc/newfs to /usr/sbin/newfs.	N
newgrp(1)	S		N
newkey(8)	S		N
nfsd(8)	S		N
nfsstat(8C)	S		N
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 /usr/bin. The default process priority for the command built into <code>cs</code> h is 4, and the default value for /usr/bin/nice is 10. The SunOS release 5.6 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 /usr/bin, 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
n1(1V) - SysV	S		N
nlsadmin(8)	C	The function of the <code>-l</code> option differs between the versions. In the SunOS release 4.x software, changing <code>addr</code> does not take effect until the next time the listener for that network is started. In the SunOS release 5.6 software, it happens immediately. In the SunOS release 4.x software, <code>addr</code> can be specified in hexadecimal notation while in the SunOS release 5.6 software it cannot. The SunOS release 4.x <code>-m</code> option is not available in the SunOS release 5.6 version. This option is used to add a new service to the list of services available through the indicated listener.	N

TABLE A-5 Commands Reference Table *(continued)*

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
nm(1)	C	The following SunOS release 4.x options are not available with the SunOS release 5.6 version: <code>-g</code> , <code>-p</code> , <code>-s</code> , and <code>-a</code> . The SunOS release 4.x and SunOS release 5.6 versions of the <code>-n</code> , <code>-o</code> and <code>-r</code> options differ.	N
nohup(1V)	C		N
nohup(1V) - SysV	S		N
nroff(1)	S		N
nslookup(8C)	S		N
nsquery(8)	S		N
nulladm(8)	S		N
od(1V)	S		N
od(1V) - SysV	S		N
old-analyze(8)	N		N
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

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
old-pti(1)	N		N
old-setkeys(1)	N		N
old-sun3cvt(1)	N		N
old-syslog(1)	N		N
old-uncompact(1)	N		N
old-vc(1)	N		N
on(1C)	S		N
overview(1)	N		N
pac(8)	N		N
pack(1V)	S		N
pack(1V) - SysV	S	With the SunOS release 4.x /usr/5bin/pack command, file names are restricted to 12 characters. In SunOS release 5.6 system software, they are restricted to {NAME_MAX} - 2. The SunOS release 5.6 pack and unpack commands are compatible with the SunOS release 4.x commands.	N
page(1)	S		N
pagesize(1)	S		S
passwd(1)	C	The -F <i>filename</i> option is not available. The -f and -s options have different meanings. The -f option forces the user to change the password at the next login. The -s option displays the password attributes for the user's login name.	N

TABLE A-5 Commands Reference Table *(continued)*

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
<code>paste(1V)</code> - SysV	S		N
<code>pax(1V)</code>	C		N
<code>paxcpio(1V)</code>	A	The <code>cpio(1)</code> and <code>pax(1)</code> commands provide similar capabilities.	N
<code>pcat(1V)</code> - SysV	S		N
<code>pdpl1(1)</code>	S		N
<code>perfmeter(1)</code>	A	This command is available in the SunOS release 5.6 software as the OpenWindows Performance Meter tool.	N
<code>pg(1V)</code> - SysV	S		N
<code>ping(8C)</code>	S		N
<code>plot(1G)</code>	N		S
<code>plottoa(1G)</code>	N		S
<code>portmap(8C)</code>	A	The <code>rpcbind(1M)</code> daemon provides similar capabilities.	N
<code>pr(1V)</code>	C		N
<code>pr(1V)</code> - SysV	S		N
<code>praudit(8)</code>	S		N
<code>prctmp(8)</code>	S		N
<code>prdaily(8)</code>	S		N
<code>printenv(1)</code>	A	The <code>env(1)</code> command provides similar capabilities.	S

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
prof(1)	C	The SunOS release 4.x <code>-v</code> option is not available with SunOS release 5.6 system software. This option suppresses all printing and produces 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 5.6 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> . The SunOS release 5.6 command 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 5.6 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 option has 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 5.6 command, it prints information in a format that reflects the new process scheduler design.	S
pstat(8)	A	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

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
pwd(1)	S		N
pwdauthd(8C)	N	Similar capabilities will be available in future releases with unbundled products. See your system vendor for information on this product.	N
quot(8)	S		N
quota(1)	S		N
quotacheck(8)	S		N
quotaoff(8)	S		N
quotaon(8)	S		N
ranlib(1)	C	The ar(1) command automatically provides similar capabilities. ranlib remains as a null script.	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 5.6 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

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
<code>rdist(1)</code>	S		N
<code>rdump(8)</code>	A	The <code>ufsdump(1M)</code> command provides similar capabilities.	N
<code>reboot(8)</code>	S		N
<code>red(1)</code>	S		N
<code>refer(1)</code>	S		N
<code>rehash(1)</code>	S		N
<code>remove_brackets(1)</code>	A	A version of this command is available with the OpenWindows Text Editor.	N
<code>renice(8)</code>	A	The <code>prIOCtl(1)</code> command provides similar capabilities.	S
<code>repquota(8)</code>	S		N
<code>reset(1)</code>	A	<code>stty</code> provides similar capabilities.	S
<code>restore(8)</code>	A	The SunOS release 5.6 command, <code>ufsrestore</code> , has been enhanced to take advantage of the end-of-media detection done by <code>ufsdump</code> .	N
<code>rev(1)</code>	N		N
<code>rexd(8C)</code>	A	<code>in.rexd</code> provides similar capabilities.	N
<code>rexecd(8C)</code>	A	<code>in.rexcd</code> provides similar capabilities.	N
<code>rfadmin(8)</code>	N	RFS is not available.	N
<code>rfpasswd(8)</code>	N	RFS is not available.	N
<code>rfstart(8)</code>	N	RFS is not available.	N
<code>rfstop(8)</code>	N	RFS is not available.	N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
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 <code>~dsusp</code> sequence for escapes on SunOS release 4.x system software is not available with the SunOS release 5.6 command. Also, the syntax for the <code>-e</code> option differs between the SunOS release 4.x and SunOS release 5.6 commands. In SunOS release 4.x system software, the syntax is <code>-ec</code> ; in SunOS release 5.6 system software, it is <code>-e c</code> .	N
rlogind(8C)	A	in <code>rlogind</code> provides similar capabilities.	N
rm(1)	S		N
rm_client(8)	A	The <code>admintool(1M)</code> utility replaces this command on SunOS release 5.6 systems.	N
rm_services(8)	A	The <code>swmtool(1M)</code> command provides similar capabilities	N
rmail(8C)	C	The SunOS release 4.x version handles remote mail received using <code>uucp(1C)</code> . It is explicitly designed for use with <code>uucp(1C)</code> and <code>sendmail(8)</code> . The SunOS release 5.6 <code>rmail</code> is a link to <code>mail(1)</code> and is a command used for reading mail.	N
rm-del(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> . The SunOS release 5.6 command silently ignores the error.	N

TABLE A-5 Commands Reference Table *(continued)*

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
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 gethostent(3) to look up all symbolic names and gateways, while the SunOS release 5.6 command uses gethostbyname(3).	N
routed(8)	A	in.routed provides similar capabilities.	N
rpc.bootparamd(8)	S		N
rpc.etherd(8C)	N	snoop(1m) obsoletes this daemon.	N
rpc.lockd(8C)	A	lockd provides similar capabilities.	N
rpc.mountd(8C)	A	mountd provides similar capabilities.	N
rpc.rexd(8C)	S		N
rpc.rquotad(8C)	S		N
rpc.rstatd(8C)	S	Now in /usr/lib/netsvc/rstat.	N
rpc.rusersd(8C)	S	Now in /usr/lib/netsvc/rusers.	N
rpc.rwalld(8C)	S	Now in /usr/lib/netsvc/rwall.	N
rpc.showfhd(8C)	A	The showfhd(1M) command provides similar capabilities.	N
rpc.sprayd(8C)	S	Now in /usr/lib/netsvc/spray.	N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
rpc.statd(8C)	S	Now in /usr/lib/netsvc/rstat.	N
rpc.user_agentd(8C)	N		N
rpc.yppasswdd(8C)	N		N
rpc.yppupdated(8C)	N		N
rpcgen(1)	S		N
rpcinfo(8)	S		N
rrestore(8)	A	The 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 5.6 command silently ignores the error.	N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
savecore(8)	S		N
sccs(1)	S		N
sccs-admin(1)	S		N
sccs-cdc(1)	S		N
sccs-comb(1)	S		N
sccs-delta(1)	S		N
sccs-get(1)	S		N
sccs-help(1)	S		N
sccs-prs(1)	S		N
sccs-prt(1)	S		N
sccs-rmdel(1)	S		N
sccs-sact(1)	S		N
sccs-sccsdiff(1)	S		N
sccs-unget(1)	S		N
sccs-val(1)	S		N
sccsdiff(1)	C		N
screenblank(1)	C	The OpenWindows <code>xset -s -600</code> command provides similar capabilities.	N
screendump(1)	N		N
screenload(1)	N		N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
script(1)	S		N
scrolldefaults(1)	C	The function of this command is now handled by the OpenWindows property window.	N
sdiff(1V) - SysV	S		N
sed(1V) - SysV	S		N
sed(1V)	C	The SunOS release 4.x /usr/5bin/sed and the SunOS release 5.6 commands do not strip initial SPACE and TAB characters from text lines.	S
selection_svc(1)	N		N
sendmail(8)	S		N
set4(8)	N		N
set_alarm(1)	N		N
setkeys(1)	N		N
setsid(8V)	N		N
setup_client(8)	N		N
setup_exec(8)	N		N
sh(1)	C	Under SunOS release 4.x system software, the behavior of the builtins echo and test depend on the relative positions of /usr/bin and /usr/5bin in the environment variable PATH. The behavior is now triggered by the relative ordering of /usr/ueb and /usr/bin.	N
shelltool(1)	C	This command is available as an OpenWindows Shell Tool.	N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
shift_lines(1)	C	An OpenWindows command is available with the OpenWindows Text Editor.	N
showfh(8C)	N		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 5.6 shutdown(1M) command. By default, the SunOS release 5.6 shutdown(1M) asks for confirmation before starting shutdown activities, while the SunOS release 4.x shutdown(8) does not ask for confirmation. In addition, the following SunOS release 4.x options are not present in the SunOS release 5.6 command: -f, -h, -k, -n, -r.	S
size(1)	C	The SunOS release 4.x command prints sizes in hexadecimal and decimal, and the file name is optional (with a.out as the default). The SunOS release 5.6 command prints them only in decimal, unless the -o or -x option is specified, and the file name is required.	N
skyversion(8)	N		N
sleep(1)	S		N
soelim(1)	S		N
sort(1V) - SysV	S		N
sort(1V)	C		N
sortbib(1)	S		N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
sparc(1)	S		N
spell(1)	C	The SunOS release 4.x <i>-h spellhist</i> option is not available with the SunOS release 5.6 command. This option places misspelled words with a user/date stamp in <i>spellhist</i> .	N
spellin(1)	S		N
spline(1G)	S		N
split(1)	S		N
spray(8C)	C	The SunOS release 4.x <i>-i delay</i> option is not available with the SunOS release 5.6 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 5.6 stty command: <i>decctlq</i> , <i>tandem</i> , <i>cbreak</i> , <i>ctlecho</i> , <i>prterase</i> , <i>crtkill</i> , <i>cols</i> , <i>tab3</i> , <i>crt</i> , <i>dec</i> , <i>term</i> .	S
stty_from_defaults(1)	N		N
su(1V) - SysV	S		N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
su(1V)	C	The SunOS release 4.x <code>-f</code> option is not supported by the SunOS release 4.x / <code>usr/5bin/su</code> or SunOS release 5.6 <code>su</code> command. This option was used for a fast <code>su</code> with <code>csch</code> .	N
sum(1V) – SysV	S		N
sum(1V)	C		S
sun(1)	S		N
sundiag(8)			N
suninstall(8)	C	The command to install SunOS release 5.6 software is still called <code>suninstall</code> , but the installation procedure has changed completely. See <i>Installation Instructions for Solaris 2.6 (SPARC Platform Edition)</i> .	N
sunview(1)	A	OpenWindows replaces SunView in SunOS release 5.6 systems.	N
sv_acquire(1)	N		N
sv_release(1)	N		N
swapon(8)	A	The <code>swap(1M)</code> command provides similar capabilities. In general, options to the SunOS release 5.6 <code>swap</code> command replace capabilities of individual swap-related commands, such as <code>swapon</code> , in SunOS release 4.x systems.	N
swin(1)	N		N
switcher(1)	N		N
symorder(1)	S		N

TABLE A-5 Commands Reference Table *(continued)*

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
sync(1)	S		N
sys-unconfig(8)	S		N
syslogd(8)	S		N
t300(1G)	N		S
t300s(1G)	N		S
t4013(1G)	N		S
t450(1G)	N		S
tabs(1V) - SysV	S		N
tail(1)	S		N
talk(1)	S		N
tar(1)	S		N
tbl(1)	S		N
tcopy(1)	S		N
tcov(1)	N	Available as an unbundled product.	N
tee(1)	S		N
tek(1G)	N		S
tektool(1)	N		N
telnet(1C)	S		N
test(1V) - SysV	S		N
test(1V)	C		S

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
textedit(1)	A	This command is available as the OpenWindows Text Edit tool.	N
textedit_filters(1)	A	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 /usr/5bin/time and the SunOS release 5.6 command. The SunOS release 4.x time 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 -f option is not available. This option attempts to force the touch 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

TABLE A-5 Commands Reference Table *(continued)*

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
trace(1)	A	The <code>truss(1)</code> command provides similar capabilities.	N
traffic(1C)	N		N
troff(1)	S		N
trpt(8C)	N		N
true(1)	S		N
tset(1)	N		S
tsort(1)	S		N
tty(1)	S		N
ttysoftcar(8)	N		N
tunefs(8)	S		N
turnacct(8)	S		N
tvconfig(8)	N		N
tzsetup(8)	N		N
u370(1)	S		N
u3b(1)	S		N
u3b15(1)	S		N
u3b2(1)	S		N
u3b5(1)	S		N
u1(1)	S		N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
umask(1)	S		N
umount(8)	C	The interface differs significantly between the two versions. In the SunOS release 5.6 command, most options are changed and must be supplied as file system-specific options.	N
umount_tfs(8)	N		N
unadv(8)	N	RFS not available.	N
uname(1)	S		N
uncompress(1)	S		N
unconfigure(8)	N		N
unexpand(1)	S		N
unget(1)	C	The versions differ in how they treat an unreadable s.file. The SunOS release 4.x version will print an error and continue if it encounters an unreadable s.file. The SunOS release 5.6 version silently ignores the error.	N
unifdef(1)	S		N
uniq(1)	S		N
units(1)	S		N
unix2dos(1)	S		N
unlink(8V)	S		N
unpack(1V) - SysV	S		N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
unpack(1V) – SysV	C	With the SunOS release 4.x /usr/5bin/pack command, file names are restricted to 12 characters. In SunOS release 5.6 system software, they are restricted to {NAME_MAX} - 2. The SunOS release 5.6 pack and unpack commands are compatible with the SunOS release 4.x commands.	
unwhiteout(1)	N		N
update(8)	A	The fsflush process provides this capability.	N
uptime(1)	A	The who -u command provides similar capabilities.	S
users(1)	A	The who -q provides similar capabilities.	S
ustar(1V)	A	The tar(1) command provides similar capabilities.	N
uuccheck(8C)	S		N
uucico(8C)	S		N
uucleanup(8C)	S		N
uucp(1C)	S		N
uudecode(1C)	S		N
uuencode(1C)	S		N
uulog(1C)	C	The -u option, which allows printing of information about work done for a specified username, is no longer supported.	N
uuname(1C)	S		N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
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		S
vswap(1)	N		N
vtroff(1)	N		N

TABLE A-5 Commands Reference Table *(continued)*

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
wwidth(1)	N		N
w(1)	S		N
wait(1)	S		N
wall(1)	S		N
wc(1)	S		N
what(1)	S		N
whatis(1)	C		N
whereis(1)	N		S
which(1)	S		N
who(1)	S		N
whoami(1)	A	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)a	N		N
xstr(1)	S		N
yacc(1)	S		N

TABLE A-5 Commands Reference Table (continued)

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
yes(1)	N		N
ypbatchupd(8C)	N		N
ypbind(8)	S	Now in /usr/lib/netsvc/yp.	N
ypcat(1)	S		N
ypinit(8)	S		N
ypmatch(1)	S		N
yppasswd(1)	S	The <code>yppasswd</code> command is still available on SunOS release 5.6 systems to access the password information on NIS servers. The equivalent command for NIS+ databases is <code>nisp passwd(1)</code> . The <code>passwd(1)</code> command can handle passwords in all supported databases (NIS, NIS+, files).	N
yppoll(8)	S		N
yppush(8)	N		N
ypserv(8)	N		N
ypset(8)	S		N
ypupdated(8C)	N		N
ypwhich(8)	S		N
ypxfr(8)	S	Now in /usr/lib/netsvc/yp.	N
ypxfrd(8)	S		N
zcat(1)	S		N

TABLE A-5 Commands Reference Table *(continued)*

SunOS Release 4.x Command	SunOS release 5.6 Status	Alternative Available and Notes	BSD
<code>zdump(8)</code>	S		N
<code>zic(8)</code>	S		N

System Calls Reference Table

This appendix contains the System Calls reference table. This table lists all SunOS release 4.x, and shows their status in the following environments: Solaris 2.6, the ABI, the SVID, SVR4, and the SunOS/BSD Source Compatibility Package.

Using the Reference Table

- If an interface is listed as “changed” (C), a brief description of differences between the SunOS release 4.x system call and the Solaris 2.6 system call is provided.
- If an interface is listed as “the same” (S), the Solaris 2.6 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 that many system calls are now available as library routines. The Notes column will show the new routine man page reference.
- If an interface has an “alternative” (A), check the Notes section for its replacement.
- If an interface is listed as “not available” (N), you cannot use that interface.
- If the interface includes `errno` values that are not supported in the standard, it is indicated with “#”. `errno` differences do not necessarily break compatibility. Note that although `EDQUOT`, `EFAULT`, and `EIO` are often not listed with ABI or SVID, these `errno` values are supported by an ABI or SVID compliant system if appropriate.

The SunOS release 4.x software offers a System V Software installation option that provides System V compatible versions of many utilities, system

calls, and library routines. The System V interfaces are included in the following tables. When referring to the System V version of a SunOS release 4.x interface, the string SysV is appended to the interface.

For complete information on all Solaris 2.6 interfaces, see the man Pages(2): System Calls.

Note - System Calls are functions. Functions in this appendix are identified by an empty set of parentheses immediately following the function name. When you see a second set of parentheses containing a number, this nomenclature identifies the associated man page section.

Examples

Below are sample table entries followed by an interpretation of the table entry.

SunOS release 4.x System Call	SunOS release 5.6 Status	Alternative Available and Notes	ABI	SVID	SVR4	BSD
mctl(2)	A	The memcntl(2) system call provides similar functionality.	A	A	A	S

The mctl() system call is not available in the ABI, SVID, SVR4, or the SunOS release 5.6 software. 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 release 5.6 Status	Alternative Available and Notes	ABI	SVID	SVR4	BSD
getsockname(2)S#		The errno value ENOBUFS used by the SunOS release 4.x getsockname() system call has been changed to ENOSR in the SVR4 and SunOS release 5.6 version.	N	N	S#	N

The `getsockname()` system call is not defined in the ABI, or SVID. The `getsockname()` call in SunOS release 5.6 and SVR4 releases is the same as the one in the SunOS release 4.x software, except the SunOS release 5.6 software sets `errno` to `ENOSR` for the error condition that previously would have set `errno` to `ENOBUFS`.

System Calls

TABLE B-1 System Calls Reference Table

SunOS Release 4.x System Call	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
WEXITSTATUS(2)	C	The <i>union wait</i> , supported in the SunOS release 4.x software for backwards compatibility, is not supported in the SVR4 and SunOS release 5.6 versions.	N	N	S	S
WIFEXITED(2)	C	The <i>union wait</i> , supported in the SunOS release 4.x software for backwards compatibility, is not supported in the SVR4 and SunOS release 5.6 versions.	N	N	C	S
WIFSIGNALED(2)	C	The <i>union wait</i> , supported in the SunOS release 4.x software for backwards compatibility, is not supported in the SVR4 and SunOS release 5.6 versions.	N	N	C	S
WIFSTOPPED(2)	C	The <i>union wait</i> , supported in the SunOS release 4.x software for backwards compatibility, is not supported in the SVR4 and SunOS release 5.6 versions.	N	N	C	S
WSTOPSIG(2)	C	The <i>union wait</i> , supported in the SunOS release 4.x software for backwards compatibility, is not supported in the SVR4 and SunOS release 5.6 versions.	N	N	C	S
WTERMSIG(2)	C	The <i>union wait</i> , supported in the SunOS release 4.x software for backwards compatibility, is not supported in the SVR4 and SunOS release 5.6 versions.	N	N	C	S

TABLE B-1 System Calls Reference Table (continued)

SunOS Release 4.x System Call	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
<code>_exit(2V)</code> — SysV	S		S	S	S	N
<code>accept(2)</code>	S	Now <code>accept(3N)</code> .	N	N	S	N
<code>access(2V)</code> — SysV	S		S	S	S	N
<code>acct(2)</code>	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 5.6, ABI, SVID, and SVR4 versions: ACOMPAT, ACORE, AXSIG. Also, the accounting record format differs between SunOS release 4.x and SunOS release 5.6, ABI, SVID, and SVR4 versions.	C#	C#	C#	N
<code>adjtime(2)</code>	S		N	S	S	N
<code>async_daemon(2)</code>	N		N	N	N	N
<code>audit(2)</code>	N		N	N	N	N
<code>auditon(2)</code>	N		N	N	N	N
<code>auditsvc(2)</code>	N		N	N	N	N
<code>bind(2)</code>	S	Now <code>bind(3N)</code> .	N	N	S	N
<code>brk(2)</code>	S		N	N	S	N
<code>chdir(2V)</code> — SysV	S		S	S	S	N

TABLE B-1 System Calls Reference Table (continued)

SunOS Release 4.x System Call	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
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 5.6, ABI, SVID, or SVR4 versions: S_IREAD (00400), S_IWRITE (00200), S_IEXEC (00100). However, the equivalent SunOS release 5.6, ABI, SVID, or SVR4 symbolic access modes S_IRUSR (00400), S_IWUSR (00200) and 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 chown() are of type int. In the SunOS release 5.6 software, ABI, SVID, and SVR4, <i>owner</i> is of type uid_t, and <i>group</i> is of type gid_t. 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 release 5.6 version, chown() changes the ownership of the file or directory referred to by the symbolic link. Use lchown(2) 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 5.6 Status	Notes	ABI	SVID	SVR4	BSD
<code>creat</code> (2V)	C#	<p>In the SunOS release 4.x software, the <i>mode</i> argument to <code>creat()</code> is of type <code>int</code>, while in SunOS release 5.6, ABI, SVID, and SVR4 versions, the <i>mode</i> argument is of type <code>mode_t</code>. Also, SunOS release 5.6, ABI, SVID, and SVR4 versions include <code><fcntl.h></code> while the SunOS release 4.x version does not.</p> <p>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 5.6, 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 5.6, ABI, SVID, and SVR4 symbolic access modes <code>S_IRUSR</code> (00400), <code>S_IWUSR</code> (00200), and <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 5.6, ABI, SVID, or SVR4 versions: <code>ENXIO</code>, <code>EOPNOTSUPP</code>.</p>	C#	C#	C#	N
<code>creat(2V)</code> — SysV	C#	<p>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 5.6, 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 5.6, ABI, SVID, and SVR4 symbolic access modes <code>S_IRUSR</code> (00400), <code>S_IWUSR</code> (00200), and <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 5.6, ABI, SVID, or SVR4 versions: <code>ENXIO</code>, <code>EOPNOTSUPP</code>.</p>	C#	C#	C#	N

TABLE B-1 System Calls Reference Table (continued)

SunOS Release 4.x System Call	SunOS release		ABI	SVID	SVR4	BSD
	5.6 Status	Notes				
dup(2V) — SysV	S		S	S	S	N
dup2(2V) — SysV	S	Now dup2(3C).	S	S	S	N
execve(2V) — SysV	S		S	S	S	N
fchdir(2V) — SysV	S		S	S	S	N
fchmod(2V) — SysV	C	The following symbolic access modes (<sys/stat.h>) are supported by the SunOS release 4.x version of fchmod(), but not by SunOS release 5.6, ABI, SVID, or SVR4 versions: S_IREAD (00400), S_IWRITE (00200), S_IEXEC (00100). However, the equivalent SunOS release 5.6, ABI, SVID, and SVR4 symbolic access modes S_IRUSR (00400), S_IWUSR (00200), and 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
fchown(2)	S		S	S	S	N
fchroot(2)	S		N	N	N	N

TABLE B-1 System Calls Reference Table (continued)

SunOS Release 4.x System Call	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
<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>-FNBIO</code> flags defined in <code><sys/file.h></code> . SunOS release 5.6, ABI, SVID, and SVR4 versions 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>-FNBIO</code> . <code>-O_NONBLOCK</code> should also be used in place of <code>-O_NDELAY</code> , which is being phased out. SunOS release 4.x <code>F_GETOWN</code> and <code>F_SETOWN</code> commands are not supported in SunOS release 5.6, ABI, SVID, or SVR4 versions.	C	C	C	N
<code>flock(2)</code>	N		N	N	N	S
<code>fork(2V)</code>	C	In the SunOS release 4.x software, <code>fork()</code> returns a value of type <code>int</code> . In SunOS release 5.6, ABI, SVID, and SVR4 versions, <code>fork()</code> returns a value of type <code>pid_t</code> . Also, SunOS release 5.6, ABI, SVID, or SVR4 versions include <code><unistd.h></code> and <code><sys/types.h></code> while the SunOS release 4.x version does not.	C	C	C	N
<code>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
<code>fstatfs(2)</code>	A	The <code>fstatvfs(2)</code> system call provides equivalent functionality.	A	A	A	S
<code>fsync(2)</code>	S		S	S	S	N

TABLE B-1 System Calls Reference Table (continued)

SunOS Release 4.x System Call	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
ftruncate(2)	S	Now ftruncate(3C).	N	N	S	N
getauid(2)	N		N	N	N	N
getdents(2)	S		N	N	S	N
getdirentries(2)	A	The getdents(2) system call provides equivalent functionality.	N	N	N	N
getdomainname(2)	A	The sysinfo(2) system call provides equivalent functionality.	N	N	N	N
getdtablesize(2)	A	Now getdtablesize(3C). The getrlimit(2) system call with the <i>resource</i> argument set to RLIMIT_NOFILE provides similar functionality.	A	A	A	S
getegid(2V)	C	In the SunOS release 4.x software, getegid() returns a value of type int. In SunOS release 5.6, ABI, SVID, and SVR4 versions, getegid() returns a value of type gid_t. Also, SunOS release 5.6, ABI, SVID, or SVR4 versions include <unistd.h> and <sys/types.h> 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 the SunOS release 4.x software, geteuid() returns a value of type int. In SunOS release 5.6, ABI, SVID, and SVR4 versions, geteuid() returns a value of type uid_t. Also, SunOS release 5.6, ABI, SVID, and SVR4 versions include <unistd.h> and <sys/types.h> while the SunOS release 4.x version does not.	C	C	C	N
geteuid(2V) — SysV	S		S	S	S	N

TABLE B-1 System Calls Reference Table (continued)

SunOS Release 4.x System Call	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
getgid(2V)	C	In the SunOS release 4.x software, <code>getgid()</code> returns a value of type <code>int</code> . In SunOS release 5.6, ABI, SVID, and SVR4 versions, <code>getgid()</code> returns a value of type <code>gid_t</code> . Also, SunOS release 5.6, 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 the SunOS release 4.x software, the <code>gidset</code> argument to <code>getgroups()</code> is of type <code>int</code> , while in SunOS release 5.6, ABI, SVID, and SVR4 versions, the <code>grouplist</code> argument is of type <code>gid_t</code> . Also, SunOS release 5.6, ABI, SVID, and SVR4 versions include <code><unistd.h></code> and <code><sys/types.h></code> while the SunOS release 4.x version does not.	C	C	C	N
getgroups(2V) — SysV	S		S	S	S	N
gethostid(2)	A	Now <code>gethostid(3C)</code> . The <code>sysinfo(2)</code> system call with the <code>command</code> argument set to <code>SI_HW_SERIAL</code> provides similar functionality.	N	N	N	S
gethostname(2)	A	Now <code>gethostname(3C)</code> . The <code>sysinfo(SI_HOSTNAME, name, namelen)</code> routine provides similar functionality.	N	N	N	S
getitimer(2)	S		N	S	S	N
getmsg(2)	S		S	S	S	N
getpagesize(2)	A	Now <code>getpagesize(3C)</code> . The <code>sysconf(3C)</code> routine provides similar functionality.	A	A	A	S

TABLE B-1 System Calls Reference Table (continued)

SunOS Release 4.x System Call	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
getpeername(2)	S#	Now getpeername(3N). The following <code>errno</code> flag is valid for the SunOS release 4.x <code>getpeername()</code> system call but is not valid in the SVR4 and SunOS release 5.6 version: <code>ENOBUFS</code> .	N	N	S#	N
getpgid(2V)	S		S	S	S	N
getpgrp(2V)	C	The SunOS release 4.x version of <code>getpgrp()</code> has an argument <code>pid</code> , and <code>getpgrp()</code> returns the process group of the process indicated by <code>pid</code> . SunOS release 5.6, 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 5.6, ABI, SVID, and SVR4 <code>getpgrp()</code> returns a value of type <code>pid_t</code> . SunOS release 5.6, 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 the SunOS release 4.x software, <code>getpid()</code> returns a value of type <code>int</code> . SunOS release 5.6, ABI, SVID, and SVR4, <code>getpid()</code> returns a value of type <code>pid_t</code> . Also, SunOS release 5.6, 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

TABLE B-1 System Calls Reference Table (continued)

SunOS Release 4.x System Call	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
getppid(2V)	C	In the SunOS release 4.x software, <code>getppid()</code> returns a value of type <code>int</code> . SunOS release 5.6, ABI, SVID, and SVR4, <code>getppid()</code> returns a value of type <code>pid_t</code> . Also, SunOS release 5.6, ABI, SVID, and SVR4 versions include <code><unistd.h></code> and <code><sys/types.h></code> while the SunOS release 4.x version does not.	C	C	C	N
getppid(2V) — SysV	S		S	S	S	N
getpriority(2)	A	Now <code>getpriority(3C)</code> . The <code>prctl(2)</code> system call provides similar functionality.	A	A	A	S
getrlimit(2)	C	In SunOS release 4.x, <code>RLIMIT_RSS</code> 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 5.6, ABI, SVID, and SVR4 versions. SunOS release 5.6, ABI, SVID, and SVR4 versions additionally support the <code>RLIMIT_AS</code> resource, the maximum amount of a process's address space that is defined (in bytes). Also, SunOS release 5.6, 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> .	C	C	C	N
getrusage(2)	A	Now <code>getusage(3C)</code> .	N	N	N	C
getsockname(2)	S#	The <code>errno</code> value <code>ENOBUFS</code> used by the SunOS release 4.x <code>getsockname()</code> system call has been changed to <code>ENOSR</code> in the SVR4 and SunOS release 5.6 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

TABLE B-1 System Calls Reference Table (continued)

SunOS Release 4.x System Call	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
getuid(2V)	C	In the SunOS release 4.x software, <code>getuid()</code> returns a value of type <code>int</code> . SunOS release 5.6, ABI, SVID, and SVR4 <code>getuid()</code> returns a value of type <code>uid_t</code> . Also, SunOS release 5.6, 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 159	C	C	C	N
kill(2V)	C	In the SunOS release 4.x software, if a signal is sent to a group of processes (as with, if <code>pid</code> 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 5.6, ABI, SVID, and SVR4 versions, the signal is sent to the sending process as well. In the SunOS release 4.x software, the <code>pid</code> argument is of type <code>int</code> , while in the SunOS release 5.6, ABI, SVID, and SVR4 versions, the <code>pid</code> argument is of type <code>pid_t</code> . Also, SunOS release 5.6, 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	Now <code>killpg(3C)</code> . 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

TABLE B-1 System Calls Reference Table (continued)

SunOS Release 4.x System Call	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
link(2V) — SysV	C	In the SunOS release 5.6, ABI, SVID, and SVR4 version of link(), 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 listen(3N).	N	N	S	N
lseek(2V) — SysV	S		S	S	S	N
lstat(2V) — SysV	S		S	S	S	N
mctl(2)	A	The memcntl(2) system call provides similar functionality.	A	A	A	S
mincore(2)	C	In the SunOS release 4.x software, argument len is of type int, while in SVR4 and SunOS release 5.6 versions, argument len is of type size_t which is defined to be unsigned int. The SunOS release 5.6 version also requires inclusion of <unistd.h>.	N	N	C	N

TABLE B-1 System Calls Reference Table (continued)

SunOS Release 4.x System Call	SunOS release 5.6 Status	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 5.6, ABI, SVID, and SVR4, the mode argument is of type <code>mode_t</code> . Also, SunOS release 5.6, 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 5.6, ABI, SVID, and SVR4 versions: <code>S_IREAD (00400)</code> , <code>S_IWRITE (00200)</code> , <code>S_IEXEC (00100)</code> . However, the equivalent SunOS release 5.6, ABI, SVID, and SVR4 symbolic access modes <code>S_IRUSR (00400)</code> , <code>S_IWUSR (00200)</code> , and <code>S_IXUSR (00100)</code> 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 5.6, ABI, SVID, and SVR4 versions: <code>S_IREAD (00400)</code> , <code>S_IWRITE (00200)</code> , <code>S_IEXEC (00100)</code> . However, the equivalent SunOS release 5.6, ABI, SVID, and SVR4 symbolic access modes <code>S_IRUSR (00400)</code> , <code>S_IWUSR (00200)</code> , and <code>S_IXUSR (00100)</code> 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 5.6 Status	Notes	ABI	SVID	SVR4	BSD
mknod(2V) — SysV	C	The mode argument to <code>mknod()</code> is of type <code>int</code> in the SunOS release 4.x software and of type <code>mode_t</code> in SunOS release 5.6, ABI, SVID, and SVR4 versions. The <i>dev</i> argument is of type <code>int</code> in the SunOS release 4.x software and of type <code>dev_t</code> in the SunOS release 5.6, ABI, SVID, or SVR4 versions. 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 5.6, ABI, SVID, and SVR4 versions: <code>S_IREAD (00400)</code> , <code>S_IWRITE (00200)</code> , <code>S_IEXEC (00100)</code> . However, the equivalent symbolic access modes <code>S_IRUSR (00400)</code> , <code>S_IWUSR (00200)</code> , and <code>S_IXUSR (00100)</code> 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
mmap(2)	C	In the SunOS release 4.x software, <code>-mmap flag</code> option value includes <code>MAP_TYPE</code> , defined in <code><sys/mman.h></code> , which is not defined in SunOS release 5.6, ABI, SVID, and SVR4 <code><sys/mman.h></code> .	C	C	C	N

TABLE B-1 System Calls Reference Table (continued)

SunOS Release 4.x System Call	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
mount(2)	C#	<p>The SunOS release 4.x version of mount() and the SunOS release 5.6, or the ABI, SVID, or SVR4 version of mount() are incompatible in a number of respects. The first argument in the SunOS release 4.x version, <i>type</i>, is the file system type name, while in SunOS release 5.6, ABI, SVID, and SVR4 versions, the first argument, <i>fs</i>, is the name of the file system. In SunOS release 5.6, 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 the SunOS release 5.6, ABI, SVID, and SVR4 version uses two parameters (five and six: <i>const char *dataptr</i> and <i>int datalen</i>). Also, SunOS release 5.6, 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</p> <p>symbolic constants for the mount() <i>flags</i> argument</p> <p>(M_NEWTYPE, M_RDONLY, M_NOSUID, M_NEWTYPE, M_GRPID, M_REMOUNT, M_NOSUB, M_MULTI) that are not defined in SunOS release 5.6, or the ABI, SVID, or SVR4 <code><sys/mount.h></code>. Instead, replace</p> <p>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 5.6, ABI, SVID, or SVR4 versions. The functionality of the following flags, defined in</p> <p><code><sys/mount.h></code>, is not supported by the SunOS release 5.6, or the ABI, SVID, or SVR4 versions: M_NOSUB, M_GRPID, M_MULTI.</p> <p>SunOS release 4.x mount() uses the following <i>errno</i></p> <p>values, which are not returned by the SunOS release 5.6, or the ABI, SVID, or SVR4 version: ENODEV, EACCES, EMFILE, ENOMEM.</p>	C#	C#	C#	N

TABLE B-1 System Calls Reference Table *(continued)*

SunOS Release 4.x System Call	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
<code>mprotect(2)</code>	S		S	S	S	N
<code>msgctl(2)</code>	S		S	S	S	N
<code>msgget(2)</code>	S		S	S	S	N
<code>msgrcv(2)</code>	S		S	S	S	N
<code>msgsnd(2)</code>	S		S	S	S	N
<code>msync(2)</code>	S		S#	S#	S	N
<code>munmap(2)</code>	S		S	S	S	N
<code>nfssvc(2)</code>	A	This interface is replaced in SunOS release 5.6 by the <code>nfssys(NFS_SVC,...)</code> ; routine.	N	N	N	N

TABLE B-1 System Calls Reference Table (continued)

SunOS Release 4.x System Call	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
<code>open(2V)</code>	C#	<p>The <i>mode</i> argument to <code>open()</code> is of type <code>int</code> in the SunOS release 4.x software and of type <code>mode_t</code> in SunOS release 5.6, ABI, SVID, and SVR4 versions. In the SunOS release 4.x software, if the <i>path</i> argument is an empty string, the kernel maps this empty pathname to '.', the current directory. In SunOS release 5.6, ABI, SVID, and SVR4 versions, if <i>path</i> points to an empty string an error results. In the SunOS release 4.x software, 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 5.6, ABI, SVID, and SVR4 versions, if the <code>O_NDELAY</code> or <code>O_NONBLOCK</code> flag is set on a call to <code>open()</code>, the corresponding flag is set for that file descriptor and subsequent reads and writes to that descriptor will not block.</p> <p>Also, SunOS release 5.6, ABI, SVID, and SVR4 versions include <code><sys/types.h></code> and <code><sys/stat.h></code> while the SunOS release 4.x version does not.</p> <p>The following <code>errno</code> value is valid for the SunOS</p> <p>4.1 version of this system call but is not returned in SunOS release 5.6, ABI, SVID, and SVR4 versions: <code>EOPNOTSUPP</code>.</p>	C#	C#	C#	N
<code>open(2V)</code> — SysV	S#	<p>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 5.6, ABI, SVID, and SVR4 versions: <code>EOPNOTSUPP</code>.</p>	S#	S#	S#	N
<code>pathconf(2V)</code> — SysV	S		S	S	S	N
<code>pipe(2V)</code> — SysV	S		S	S	S	N
<code>poll(2)</code>	S		S	S	S	N

TABLE B-1 System Calls Reference Table (continued)

SunOS Release 4.x System Call	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
profil(2)	S		S	S	S	N
ptrace(2)	C#	<p>The optional <i>addr2</i> argument to the SunOS release 4.x software <code>ptrace()</code> system call is not supported by the SunOS release 5.6 routine. The request argument to <code>ptrace()</code> is of type <code>enum ptracereq</code> in the SunOS release 4.x software and of type <code>int</code> in the SunOS release 5.6 version.</p> <p>The <i>pid</i> argument to <code>ptrace()</code> is of type <code>int</code> in the SunOS release 4.x software and of type <code>pid_t</code> in the SunOS release 5.6 version. Also, the SunOS release 5.6 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>.</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 release 5.6 version: <code>EPERM</code>.</p> <p>See "ptrace() Request Values" on page 162 for information on valid <i>request</i> values.</p>	C#	C#	C#	N
putmsg(2)	S		S	S	S	N
quotactl(2)	A	The <code>Q_QUOTACTL</code> <code>ioctl</code> system call provides similar functionality.	A	A	A	N
read(2V)	C#	The following <code>errno</code> flags are valid for the SunOS release 4.x version of this system call but are not valid in SunOS release 5.6, 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 5.6 Status	Notes	ABI	SVID	SVR4	BSD
read(2V) — SysV	C#	<p>The <i>nbyte</i> argument to <code>read()</code> is of type <code>int</code> in the SunOS release 4.x software and of type <code>unsigned</code> in the SunOS release 5.6 version.</p> <p>The SunOS release 5.6 <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 the SunOS release 4.x <code>read()</code> routine.</p> <p>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 release 5.6 version: <code>EISDIR</code>, <code>EWOULDBLOCK</code>.</p>	C#	C#	C#	N
readlink(2)	S		S	S	S	N
readv(2V)	C#	<p>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 5.6, ABI, SVID, and SVR4 versions: <code>EISDIR</code>, <code>EWOULDBLOCK</code>.</p>	C#	C#	C#	N
readv(2V) — SysV	C#	<p>SunOS release 4.x and SunOS release 5.6, or the SVID or SVR4 <code>iovec</code> structures (defined in <code><sys/uio.h></code>) differ slightly. The SunOS release 4.x <code>iovec</code> <code>iov_len</code> field is defined as <code>integer</code>, while SunOS release 5.6 or the SVID or SVR4 <code>iov_len</code> is defined as <code>unsigned</code>. SunOS release 5.6 or the SVID or SVR4 <code>readv()</code> system call does not support BSD 4.2 style non-blocking I/O.</p>	C#	C#	C#	N
reboot(2)	A	<p>Now <code>reboot(3C)</code>. The <code>uadmin(2)</code> system call provides similar functionality.</p>	N	N	N	S
recv(2)	S	<p>Now <code>recv(3N)</code>.</p>	N	N	S	N

TABLE B-1 System Calls Reference Table (continued)

SunOS Release 4.x System Call	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
recvfrom(2)	S	Now recvfrom(3N).	N	N	S	N
recvmsg(2)	S	Now recvmsg(3N).	N	N	S	N
rename(2V) — SysV	S#	The SunOS release 5.6, ABI, SVID, and SVR4 versions include <unistd.h> 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 5.6, ABI, SVID, and SVR4 versions: <code>ENOTEMPTY</code> . SunOS release 5.6, ABI, SVID, and SVR4 versions set <code>errno</code> to flag <code>EEXIST</code> instead.	S#	S#	S#	N
rmdir(2V) — SysV	S#	The SunOS release 5.6, ABI, SVID, and SVR4 versions include <unistd.h> 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 5.6, ABI, SVID, and SVR4 versions: <code>ENOTEMPTY</code> . SunOS release 5.6, ABI, SVID, and SVR4 versions set <code>errno</code> to flag <code>EEXIST</code> instead.	S#	S#	S#	N
sbrk(2)	S		N	N	S	N
select(2)	S	Now select(3C).	N	N	S	N
semctl(2)	S		S	S	S	N
semget(2)	S		S	S	S	N
semop(2)	S		S	S	S	N
send(2)	S#	Now send(3N). The following <code>errno</code> flag is valid for SunOS release 4.x <code>send(2)</code> system calls but is not valid in the SVR4 and SunOS release 5.6: <code>ENOBUFS</code> .	N	N	S#	N

TABLE B-1 System Calls Reference Table (continued)

SunOS Release 4.x System Call	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
sendmsg(2)	S#	Now sendmsg(3N). The following <code>errno</code> flag is valid for SunOS release 4.x <code>sendmsg(2)</code> system calls but is not valid in the SVR4 and SunOS release 5.6: <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 <code>sendto(2)</code> system calls but is not valid in the SVR4 and SunOS release 5.6: <code>ENOBUFS</code> .	N	N	S#	N
setaudit(2)	N		N	N	N	N
setaudit(2)	N		N	N	N	N
setdomainname(2)	A	The <code>sysinfo(2)</code> system call provides similar functionality.	N	N	N	N
setgroups(2V)	C	In the SunOS release 4.x software, the <code>gidset</code> argument is of type <code>int</code> , while in SunOS release 5.6, ABI, SVID, and SVR4 versions, the <code>grouplist</code> argument is of type <code>gid_t</code> . Also, SunOS release 5.6, ABI, SVID, and SVR4 versions include <code><unistd.h></code> and <code><sys/types.h></code> while the SunOS release 4.x version does not.	C	C	C	N
setgroups(2V) — SysV	S		S	S	S	N
sethostname(2)	A	Now <code>sethostname(3C)</code> . 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 5.6 Status	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 release 5.6 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 release 5.6 <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 5.6, <code>setpgrp()</code> returns a value of type <code>pid_t</code> . The SunOS release 5.6 version includes <code><unistd.h></code> and <code><sys/types.h></code> while the SunOS release 4.x version does not. 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 5.6, ABI, SVID, and SVR4 versions: <code>EACCES</code> , <code>EINVAL</code> , <code>ESRCH</code> .	C#	C#	C#	N
setpgrp(2V) — SysV	S	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 5.6, ABI, SVID, and SVR4 versions: <code>EACCES</code> , <code>EINVAL</code> , <code>ESRCH</code> .	S	S	S	N
setpriority(2)	A	Now <code>setpriority(3C)</code> . The <code>pricntl(2)</code> system call provides similar functionality.	A	A	A	S
setregid(2)	S	Now <code>setregid(3C)</code> .	N	N	N	C
setreuid(2)	S	Now <code>setreuid(3C)</code> .	N	N	N	C
setrlimit(2)	C	Now <code>setrlimit(3C)</code> .	C	C	C	N
setsid(2V) — SysV	S		S	S	S	N

TABLE B-1 System Calls Reference Table (continued)

SunOS Release 4.x System Call	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
setsockopt(2)	S	Now setsockopt(3N).	N	N	S	N
settimeofday(2)	S	Now settimeofday(3C).	N	S	S	S
setuseraudit(2)	N		N	N	N	N
sgetl(2)	S	Now xdr_simple(3N).	N	S	S	N
shmat(2)	S		S	S	S	N
shmctl(2)	S		S	S	S	N
shmdt(2)	S		S	S	S	N
shmget(2)	S		S	S	S	N
shutdown(2)	S	Now shutdown(3N).	N	N	S	N
sigaction(2)	C	There is a flag in the Solaris 2.6 version, SA_RESTART, that 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
sigblock(2)	A	The sigprocmask(2) system call with the how argument set to SIG_BLOCK provides similar functionality.	A	A	A	S
sigmask(2)	A	The sigsetops(3C) routines provide similar functionality.	A	A	A	S
sigpause(2V) — SysV	S	The SunOS release 4.x sigpause() system call assigns its argument (sigmask) to the set of masked signals while the ABI and SVID versions of sigpause remove its argument (sig) from the calling process's signal mask. The SVR4 and SunOS release 5.6 sigpause() is compatible with SunOS release 4.x sigpause(2).	C	C	S	S

TABLE B-1 System Calls Reference Table (continued)

SunOS Release 4.x System Call	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
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 how argument set to SIG_SETMASK provides similar functionality.	A	A	A	S
sigstack(2)	A	Now sigstack(3C). The sigaltstack(2) system call provides similar functionality.	A	A	A	S
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). The SunOS release 4.x PF_IMPIPKNK is a supported <i>domain</i> , while in SVR4 and SunOS release 5.6 software PF_IMPIPKNK is not supported. The following <i>errno</i> flags are valid for the SunOS release 4.x socket() system call but are not valid in the SVR4 and SunOS release 5.6 version: ENOBUFS, EPROTOTYPE.	N	N	C#	N
socketpair(2)	S	Now socketpair(3N).	N	N	S	N
sput1(2)	S	Now xdr_simple(3N).	N	S	S	N
stat(2V) — SysV	S		S	S	S	N
statfs(2)	A	The statvfs(2) system call provides similar functionality.	A	A	A	N
swapon(2)	A	The swapctl(2) system call provides similar functionality.	N	N	N	N

TABLE B-1 System Calls Reference Table (continued)

SunOS Release 4.x System Call	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
symlink(2)	S		S	S	S	N
sync(2)	S		S	S	S	N
syscall(2)	N		N	N	N	S
sysconf(2V) — SysV	S	Now sysconf(3C).	S	S	S	N
tell(2V) — SysV	S		S	N	S	N
truncate(2)	S	Now truncate(3C).	N	N	S	N
umask(2V) — SysV	C	The following symbolic access modes (<sys/stat.h>) are supported by the SunOS release 4.x version of umask(), but not by SunOS release 5.6, ABI, SVID, and SVR4 versions: S_IREAD (00400), S_IWRITE (00200), S_IEXEC (00100). However, the equivalent SunOS release 5.6, ABI, SVID, and SVR4 symbolic access modes, S_IRUSR (00400), S_IWUSR (00200), and S_IXUSR (00100) do have the same definitions, are defined in SunOS release 4.x <sys/stat.h>, and thus should be used.	C	C	C	N
umount(2V) — SysV	S		S	S	S	N
uname(2V) — SysV	S		S	S	S	N
unlink(2V) — SysV	S		S	S	S	N
unmount(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

TABLE B-1 System Calls Reference Table (continued)

SunOS Release 4.x System Call	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
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 5.6, ABI, SVID, and SVR4, wait() returns a value of type pid_t. Also, SunOS release 5.6, ABI, SVID, and SVR4 versions include <sys/types.h> 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 5.6, 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 5.6, ABI, SVID, and SVR4, the wait() system call returns prematurely if a signal is received.	C	C	C	N
wait(2V) — SysV	C	The <i>union wait</i> , supported in the SunOS release 4.x software for backwards compatibility, is not supported in SunOS release 5.6, ABI, SVID, and SVR4 versions. The 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 the SunOS release 5.6, ABI, SVID, and SVR4 versions, the wait(2) function will return prematurely if a signal is received.	C	C	C	N
wait3(2V)	A	Now wait3(3C). 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 5.6 Status	Notes	ABI	SVID	SVR4	BSD
wait4(2V)	A	Now wait4(3C). The wait(2) and waitpid(2) system calls provide similar functionality.	A	A	A	S
waitpid(2V)	C	SunOS release 4.x waitpid() returns a value of type int. In the SunOS release 5.6, ABI, SVID, and SVR4 versions, waitpid() returns a value of type pid_t. The <i>pid</i> argument to waitpid() is of type int in the SunOS release 4.x software and of type pid_t in the SunOS release 5.6, ABI, SVID, and SVR4 versions. Also, the SunOS release 5.6, ABI, SVID, and SVR4 versions include <sys/types.h> while the SunOS release 4.x version does not. The <i>union wait</i> , supported in the SunOS release 4.x software for backwards compatibility, is not supported in SunOS release 5.6, ABI, SVID, and SVR4 versions. The SunOS release 4.x waitpid() 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 5.6, ABI, SVID, and SVR4 versions, 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 the SunOS release 4.x software for backwards compatibility, is not supported in SunOS release 5.6, ABI, SVID, and SVR4 versions. The SunOS release 4.x waitpid(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 5.6, ABI, SVID, and SVR4 versions, the waitpid(2) function will return prematurely if a signal is received.	C	C	C	N

TABLE B-1 System Calls Reference Table (continued)

SunOS Release 4.x System Call	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
<code>write(2V)</code>	C#	<p>In the SunOS release 4.x software, if the object that the descriptor refers to is marked for non-blocking I/O, using the <code>FIONBIO</code> request to <code>ioctl()</code>, or using <code>fcntl()</code> to set the <code>FNDELAY</code> or <code>O_NDELAY</code> flag, <code>write()</code> returns -1 and sets <code>errno</code> to <code>EWOULDBLOCK</code>.</p> <p>In the SunOS release 5.6 software, on a <code>write()</code> to a regular file, if <code>O_NDELAY</code> or <code>O_NONBLOCK</code> is set, <code>write()</code> returns -1 and sets <code>errno</code> to <code>EAGAIN</code>.</p> <p>On <code>write()</code> requests to a pipe or FIFO with <code>O_NONBLOCK</code> or <code>O_NDELAY</code> set, <code>write()</code> does not block the process. If some data can be written without blocking the process, <code>write()</code> writes what it can and returns the number of bytes written; otherwise, when <code>O_NONBLOCK</code> is set, it returns -1 and sets <code>errno</code> to <code>EAGAIN</code> and when <code>O_NDELAY</code> is set, it returns 0.</p> <p>With <code>O_NDELAY</code> set, <code>write()</code> requests for <code>{PIPE_BUF}</code> or fewer bytes either succeed completely and return <i>nbytes</i>, or return 0. A <code>write()</code> request for greater than <code>{PIPE_BUF}</code> bytes either transfers what it can and returns the number of bytes written, or transfers no data and returns 0. Also, if a request is greater than <code>{PIPE_BUF}</code> bytes and all data previously written to the pipe has been read, <code>write()</code> transfers at least <code>{PIPE_BUF}</code> bytes.</p> <p>The SunOS release 5.6 <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 release 5.6 version: <code>EWOULDBLOCK</code>.</p>	C#	C#	C#	N

TABLE B-1 System Calls Reference Table (continued)

SunOS Release 4.x System Call	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
<code>write(2V)</code> — SysV	C#	The SunOS release 5.6, ABI, SVID, and SVR4 versions of <code>write()</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 SunOS release 5.6, ABI, SVID, and SVR4 versions: <code>EWOULDBLOCK</code> .	C#	C#	C#	N
<code>writew(2V)</code>	C#	SunOS release 5.6, ABI, SVID, and SVR4 versions of <code>writew()</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 5.6, or the ABI, SVID, or SVR4 version: <code>EWOULDBLOCK</code> .	C#	C#	C#	N

Library Routines Reference Table

This appendix contains the Library Routine reference table. This table lists all SunOS release 4.x library routines and shows their status in the Solaris 2.6, the ABI, the SVID, SVR4, and the SunOS/BSD Source Compatibility Package environments.

Using the Reference Table

- If an interface is listed as “changed” (C), a brief description of differences between the SunOS release 4.x and the Solaris 2.6 routine is provided.
- If an interface is listed as “the same” (S), the Solaris 2.6 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 5.6 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 2.6 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 5.6 Status	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 the SunOS release 5.6 and SVR4 versions.	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 5.6 Status	Alternative Available and Notes	ABI	SVID	SVR4	BSD
<code>putpwent(3)</code>	S		S	S	S	N

The SunOS release 4.x `putpwent()` routine and the SunOS release 5.6 routine are the same. Applications that use this routine will behave as they did in the SunOS release 4.x software.

Library Routines

TABLE C-1 Library Routines Reference Table

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
<code>_crypt(3)</code>	C	The <code>crypt(3C)</code> routine provides similar functionality.	N	A	A	N
<code>_longjmp(3)</code>	C	Now <code>_longjmp(3C)</code> . The <code>siglongjmp(3)</code> routine provides similar functionality.	A	A	A	S
<code>_setjmp(3)</code>	C	Now <code>setjmp(3C)</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 software, <code>HUGE</code> is defined in <code><math.h></code> as <code>infinity(3M)</code> , which produces IEEE Infinity. In SunOS release 5.6, SVID, or SVR4 versions, <code>HUGE</code> is defined in <code><math.h></code> as a machine-dependent constant.	N	C	C	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
HUGE_VAL(3M)	C	In the SunOS release 4.x software, HUGE_VAL is defined in <math.h> as infinity(3M), which produces IEEE Infinity. In the SunOS release 5.6, SVID, or SVR4 versions, HUGE_VAL is defined in <math.h> as a machine-dependent constant.	N	C	C	N
MONITOR(3L)	N		N	N	N	N
MSG_RECVALL(3L)	N		N	N	N	N
SAMECV(3L)	N		N	N	N	N
SAMEMON(3L)	N		N	N	N	N
SAMETHREAD(3L)	N		N	N	N	N
STKTOP(3L)	N		N	N	N	N
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 software, if the absolute value of the argument of acos() is greater than 1, NaN is returned with an EDOM error and a DOMAIN math err. The SunOS release 5.6, the SVID, or SVR4 versions return zero with an EDOM error and a DOMAIN math err.	N	C	C	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
acosh(3M)	S		N	S	S	N
addch(3V) - SysV	S		N	S	S	N
addexportent(3)	A	The /etc/dfs/sharetab file replaces /etc/exports. Refer to the share(1M), unshare(1M), and sharetab(4) man pages for more information.	N	N	N	N
addmntent(3)	A	The putmntent() routine provides similar functionality. Refer to getmntent(3C).	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
alphasort(3)	N		N	N	N	S

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
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 software, if the absolute value of the argument of asin() is greater than 1, NaN is returned with an EDOM error and a DOMAIN math err. The SunOS release 5.6, 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 the SunOS release 5.6, ABI, SVID, or SVR4 versions call abort(3C).	C	C	C	N
assert(3V) - SysV	S		S	S	S	N
atan(3M)	S		N	S	S	N
atan2(3M)	C	The SunOS release 5.6, the SVID, or SVR4 version of 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

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
atanh(3M)	S		N	S	S	N
atof(3)	C	See strtod(3).	C	C	C	N
atoi(3)	S		S	S	S	N
atol(3)	S		S	S	S	N
atrtroff(3V) - SysV	S		N	S	S	N
attron(3V) - SysV	S		N	S	S	N
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)	A	This routine is still available, but is superseded by authdes_seccreate(3N) in SunOS release 5.6, or the ABI, SVID, or SVR4.	A	A	A	N
authdes_getucrcd(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)	A	This routine is still available, but is superseded by authsys_seccreate(3N).	A	A	A	N
authunix_create_default(3N)	A	This routine is still available, but is superseded by authsys_create_default(3N).	A	A	A	N
baudrate(3V) - SysV	S		N	S	S	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
bcmp(3)	S	Now bcmp(3C).	A	A	A	S
bcopy(3)	S	Now bcopy(3C).	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
box(3V)	C	<p>The SunOS 4.x version of box() is a function while the SunOS release 5.6, or the SVID or SVR4 version of box(win, verch, horch) is a macro that calls</p> <pre>wborder(win, verch, verch, horch, horch, 0, 0, 0, 0).</pre> <p>Default values defined in < curses.h> in the SunOS release 5.6, or the SVID or SVR4 environment— ACS_ULCORNER, ACS_URCORNER, ACS_BLCORNER, and ACS_BRCORNER, 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 verch and horch in the SunOS 4.x software is char, while in SunOS release 5.6, or the SVID or SVR4 versions they are ch type.</p>	N	C	C	S
box(3V) - SysV	S		N	S	S	N
bsearch(3)	S		S	S	S	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
byteorder(3N)	S		N	N	S	N
bzero(3)	S	Now bzero(3C).	A	A	A	S
calloc(3)	S		S	S	S	N
callrpc(3N)	A	This routine is still available, but is superseded by rpc_call(3N).	N	N	S	N
catclose(3C)	S		S	S	S	N
catgetmsg(3C)	A	In the SunOS release 5.6, or the ABI, SVID, or SVR4 environment, use catgets(3C) followed by strncpy(3) to copy the catalog message from the internal buffer area to a program buffer.	A	A	A	N
catgets(3C)	S		S	S	S	N
catopen(3C)	S		S	S	S	N
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

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
<code>cfree(3)</code>	A	This routine is replaced by <code>void free(void*ptr)</code> . Refer to <code>malloc(3C)</code> man page.	A	A	A	N
<code>cfsetispeed(3V)</code>	S		S	S	S	N
<code>cfsetospeed(3V)</code>	S		S	S	S	N
<code>circle(3X)</code>	S		N	N	N	N
<code>clear(3V) - SysV</code>	S		N	S	S	S
<code>clearerr(3V) - SysV</code>	S		S	S	S	N
<code>clearok(3V) - SysV</code>	S		N	S	S	S
<code>clnt_broadcast(3N)</code>	S	This routine is still available, but is superseded by <code>rpc_broadcast(3N)</code> .	A	A	A	N
<code>clnt_call(3N)</code>	S		S	S	S	N
<code>clnt_control(3N)</code>	S		S	S	S	N
<code>clnt_create(3N)</code>	S		S	S	S	N
<code>clnt_destroy(3N)</code>	S		S	S	S	N
<code>clnt_freeres(3N)</code>	S		S	S	S	N
<code>clnt_geterr(3N)</code>	S		S	S	S	N
<code>clnt_pcreateerror(3N)</code>	S		S	S	S	N
<code>clnt_perrno(3N)</code>	S		S	S	S	N
<code>clnt_perror(3N)</code>	S		S	S	S	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
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 the SunOS release 5.6 and SVR4 software.	A	A	S	N
clnttcp_create(3N)	S	This routine is still available, but is superseded by the <code>clnt_create(3N)</code> , <code>clnt_tli_create(3N)</code> , and <code>clnt_vc_create(3N)</code> routines in SunOS release 5.6, 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.	N	N	S	
clntudp_create(3N)	S	This routine is still available, but is superseded by the <code>clnt_create(3N)</code> , <code>clnt_tli_create(3N)</code> , and <code>clnt_dg_create(3N)</code> routines .	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

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
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
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	<p>For arguments that are much lower than zero, the SunOS release 5.6, or the SVID or SVR4 version of these routines returns 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.</p> <p>The variable <code>fp_pi</code> defined in <code><math.h></code> allows changing of the precision at runtime. The error exceptions occur in the IEEE 754 spirit for both versions.</p>	N	C	C	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
cosh(3M)	S		N	S	S	N
crmode(3X)	A	This routine is replaced by <code>cbreak()</code> . See <code>curl_inopts(3X)</code> .	A	A	A	N
crypt(3)	C	In the SunOS 4.x version, the first two characters of the <i>salt</i> argument are interpreted and checked for (<code>##and#\$</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 <i>salt</i> is returned from <code>crypt</code> . Otherwise, <code>NULL</code> is returned. In the SunOS release 5.6, 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 5.6 Status	Notes	ABI	SVID	SVR4	BSD
ctime(3V)	C	<p>The SunOS 4.x <code>tm</code> structure contains two fields not present in the SunOS release 5.6, or the ABI, SVID, or SVR4 <code>tm</code> structure: <code>tm_zone</code> and <code>tm_gmtoff</code>. Instead, the SunOS release 5.6, 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, the SunOS release 5.6, 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 the SunOS 4.x and the SunOS release 5.6, or the ABI, SVID, or SVR4 versions. In the SunOS release 4.x version, <code>TZ</code> contains the pathname of <code>tzfile-format</code> file from which to read the time conversion information. In the SunOS release 5.6, or the ABI, SVID, or SVR4 versions, <code>TZ</code> itself contains the time conversion information (of different format than the <code>tzfile-format</code>).</p>	C	C	C	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
<code>curs_set(3V)</code> - SysV	S		N	S	S	N
<code>cuserid(3V)</code>	S		S	S	S	N
<code>cv_broadcast(3L)</code>	N		N	N	N	N
<code>cv_create(3L)</code>	N		N	N	N	N
<code>cv_destroy(3L)</code>	N		N	N	N	N
<code>cv_enumerate(3L)</code>	N		N	N	N	N
<code>cv_notify(3L)</code>	N		N	N	N	N
<code>cv_send(3L)</code>	N		N	N	N	N
<code>cv_wait(3L)</code>	N		N	N	N	N
<code>cv_waiters(3L)</code>	N		N	N	N	N
<code>dbm_clearerr(3)</code>	S		N	N	N	N
<code>dbm_close(3X)</code>	S	The <code>dbm_close(3)</code> routine provides similar functionality.	N	N	N	N
<code>dbm_delete(3)</code>	S	The <code>dbm_delete(3)</code> routine provides similar functionality.	N	N	N	N
<code>dbm_error(3)</code>	S		N	N	N	N
<code>dbm_fetch(3)</code>	S		N	N	N	N
<code>dbm_firstkey(3)</code>	S		N	N	N	N
<code>dbm_nextkey(3)</code>	S		N	N	N	N
<code>dbm_open(3)</code>	S		N	N	N	N

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
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
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

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
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 5.6 Status	Notes	ABI	SVID	SVR4	BSD
endexportent(3)	A	The /etc/dfs/sharetab file replaces /etc/exports. Refer to share(1M), unshare(1M), and sharetab(4) for more information.	A	A	A	N
endfsent(3)	A	This routine is replaced by fclose(3).	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 fclose(3).	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 ttymon(1) and ttydefs(4) for information about SunOS release 5.6 tty system.	N	N	N	N
endusershell(3)	S		N	N	N	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
endwin(3V)	C	The SunOS 4.x version of endwin() return value is undefined, while the SunOS release 5.6, 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
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

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
<code>exc_notify(3L)</code>	N		N	N	N	N
<code>exc_on_exit(3L)</code>	N		N	N	N	N
<code>exc_raise(3L)</code>	N		N	N	N	N
<code>exc_unhandle(3L)</code>	N		N	N	N	N
<code>exc_uniqpatt(3L)</code>	N		N	N	N	N
<code>execl(3V)</code>	C		C	C	C	N
<code>execl(3V) - SysV</code>	S		S	S	S	N
<code>execle(3V)</code>	C		C	C	C	N
<code>execle(3V) - SysV</code>	S		S	S	S	N
<code>execlp(3V)</code>	C		C	C	C	N
<code>execlp(3V) - SysV</code>	S		S	S	S	N
<code>execv(3V)</code>	C		C	C	C	N
<code>execv(3V) - SysV</code>	S		S	S	S	N
<code>execvp(3V)</code>	C		C	C	C	N
<code>execvp(3V) - SysV</code>	S		S	S	S	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
exit(3)	C	Both the SunOS 4.x and SunOS release 5.6, or the ABI, SVID, or SVR4 exit() routines do additional processing before the process exits. The SunOS 4.x exit() calls all functions registered by the on_exit(3) routine while SunOS release 5.6, or the ABI, SVID, or SVR4 exit() calls all functions registered by the atexit() routine. If no functions have been added using the on_exit(3) routine, then the SunOS 4.x and SunOS release 5.6, or the ABI, SVID, or SVR4 versions of exit() are compatible.	C	C	C	N
exp(3M)	C	In the SunOS release 5.6, or the SVID or SVR4 version, exp() 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 version, since HUGE is defined as +Infinity, exp(HUGE) and exp(-HUGE) do not overflow or underflow, hence no errno is produced. The SunOS release 5.6, or the SVID or SVR4 version sets errno to ERANGE.	N	C	C	N
exp10(3M)	N		N	N	N	N
exp2(3M)	N		N	N	N	N

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
expm1(3M)	N		N	N	N	N
exportent(3)	A	The /etc/dfs/sharetab file replaces /etc/exports. Refer to share(1M), unshare(1M), and sharetab(4) man pages for more information.	A	A	A	N
extended_to_decimal(3)	S		N	N	N	N
fabs(3M)	S		N	S	S	N
fclose(3S)	S		S	S	S	N
fconvert(3)	S		N	N	N	N
fcvt(3)	S		N	N	S	N
fdopen(3V)	S		S	S	S	N
feof(3V)	S		S	S	S	N
ferror(3V)	S		S	S	S	N
fetch(3X)	A	This routine is replaced by dbm_fetch(3) in the SunOS release 5.6 software.	N	N	N	S
fflush(3S)	S		S	S	S	N
ffs(3)	S		N	N	S	N
fgetc(3V)	S		S	S	S	N
fgetgraent(3)	N		N	N	N	N
fgetgrent(3V)	S		N	S	S	N

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
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)	A	This routine is replaced by dbm_firstkey(3) in the SunOS release 5.6 software.	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

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
fmod(3M)	C	In the SunOS release 5.6, or the SVID or SVR4 version, <code>fmod(x, 0.0)</code> returns <code>x</code> and sets <code>errno</code> to <code>EDOM</code> . In the SunOS 4.x version, the same call returns <code>NaN</code> in conformance with 4.3 BSD and in the spirit of ANSI/IEEE Std 754-1985.	N	C	C	N
fopen(3V)	S		S	S	S	S
fp_class(3M)	N		N	N	N	N
fprintf(3V)	S		S	S	S	S
fputc(3S)	S		S	S	S	N
fputs(3S)	S		S	S	S	N
fread(3S)	S		S	S	S	N
free(3)	S		S	S	S	N
freopen(3V)	S		S	S	S	S
frexp(3M)	S		N	S	S	N
fscanf(3V)	S		S	S	S	N
fseek(3S)	S		S	S	S	N
ftell(3S)	S		S	S	S	N
ftime(3V)	S	Now <code>ftime(3C)</code> .	A	A	A	S
ftok(3)	S		S	S	S	N
ftw(3)	S		S	S	S	N

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
func_to_decimal(3)	N		N	N	N	N
fwrite(3S)	S		S	S	S	N
gamma(3M)	S		N	S	S	N
garbagedlines(3V) - SysV	S		N	N	N	N
gcd(3X)	S		N	N	N	N
gconvert(3)	S		N	N	N	N
gcvt(3)	S		N	N	S	N
getacdir(3)	N		N	N	N	N
getacflg(3)	N		N	N	N	N
getacinfo(3)	N		N	N	N	N
getacmin(3)	N		N	N	N	N
getauditflagsbin(3)	N		N	N	N	N
getauditflagschar(3)	N		N	N	N	N
getbegyx(3V) - SysV	S		N	S	S	N
getc(3V)	S		S	S	S	N
getcap(3X)	N		N	N	N	S

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
getch(3V)	C	In the SunOS release 5.6, or the SVID or SVR4 software, 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 software, 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 release 5.6 getcwd() routine is compatible with the SunOS 4.x version 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. Refer to share(1M), unshare(1M), and sharetab(4) man pages for more information.	A	A	A	N
getexportopt(3)	A	The /etc/dfs/sharetab file replaces /etc/exports. Refer to share(1M), unshare(1M), and sharetab(4) man pages for more information.	A	A	A	N

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
getfauditflags(3)	N		N	N	N	N
getfsent(3)	A	This routine is replaced by getvfsent(3).	N	N	N	N
getfsfile(3)	A	This routine is replaced by getvfsfile(3).	N	N	N	N
getfsspec(3)	A	This routine is replaced by getvfsfile(3).	N	N	N	N
getfstype(3)	A	This routine is replaced by getvfsany(3).	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

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
getmntent(3)	C	The SunOS 4.x getmntent() routine and the SunOS release 5.6, SVID, or SVR4 getmntent() routine are incompatible. The SunOS 4.x getmntent() returns a pointer to an object of type mntent while SunOS release 5.6, or the SVID or SVR4 getmntent() returns int. Additionally, SunOS release 5.6, or the SVID or SVR4 getmntent() uses a different incompatible structure type (mnttab) to return the file entry type. Additionally, null pointers are returned for corresponding '.' entries in /etc/vfstab.	N	C	C	N
get_myaddress(3N)	S	This routine is still available, but is superseded by netdir_getbyname(3N).	S	N	S	N
getnetbyaddr(3N)	S		N	N	S	N
getnetbyname(3N)	S		N	N	S	N
getnetent(3N)	S		N	N	S	N
getnetgrent(3N)	N		N	N	N	N
getnetname(3N)	S		S	S	S	N
getopt(3)	S		S	S	S	N
getpass(3V)	S		S	S	S	N
getprotobyname(3N)	S		N	N	S	N
getprotobynumber(3N)	S		N	N	S	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
getprotoent(3N)	S		N	N	S	N
getpublickey(3R)	S		S	S	S	N
getpw(3)	S		N	N	S	N
getpwaent(3)	N		N	N	N	N
getpwanam(3)	N		N	N	N	N
getpwent(3V)	S		S	S	S	N
getpwnam(3V)	S		S	S	S	N
getpwuid(3V)	S		S	S	S	N
getrpcbyname(3N)	S		S	S	S	N
getrpcbynumber(3N)	S		S	S	S	N
getrpcport(3N)	S		S	S	S	N
getrpcport(3R)	A	pmap_getport() can be used to get the same result.	N	N	N	N
gets(3S)	S		S	S	S	N
getsecretkey(3R)	S		S	S	S	N
getservbyname(3N)	S		N	N	S	N
getservbyport(3N)	S		N	N	S	N
getservent(3N)	S		N	N	S	N
getstr(3V) - SysV	C		N	C	C	S
getsubopt(3)	S		S	S	S	N

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
<code>getsyx(3V)</code> – SysV	S		N	S	S	N
<code>gettext(3)</code>	C	In SunOs 5.4 <code>gettext(3)</code> searches NLSPATH first for the location of the LC_MESSAGES directory.	N	N	N	N
<code>gettmode(3V)</code>	C	The SunOS release 5.6 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 the most commonly used routines as macros for increased performance.	N	N	N	S
<code>gettmode(3V)</code> – SysV	S		N	N	N	N
<code>getttyent(3)</code>	A	Refer to <code>ttymon(1)</code> and <code>ttydefs(4)</code> for information about the SunOS release 5.6 tty system.	N	N	N	N
<code>getttynam(3)</code>	A	Refer to <code>ttymon(1)</code> and <code>ttydefs(4)</code> for information about the SunOS release 5.6 tty system.	N	N	N	N
<code>getusershell(3)</code>	S		N	N	N	N
<code>getw(3V)</code>	S		S	S	S	N
<code>getwd(3)</code>	S	Now <code>getwd(3C)</code> .	A	A	A	S
<code>getyx(3V)</code> – SysV	S		N	S	S	S
<code>gmtime(3V)</code>	C	See <code>ctime(3V)</code> .	C	C	C	N
<code>grpauth(3)</code>	N		N	N	N	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	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 5.6, or the SVID, or SVR4 version of <code>idlok()</code> sets a flag that controls whether 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

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
ieee_handler(3M)	N		N	N	N	N
ieee_retrospective (3M)	N		N	N	N	N
ilogb(3M)	N		N	N	N	N
inch(3V) - SysV	S		N	S	S	S
index(3)	S	Now is index(3C).	A	A	A	S
inet_lnaof(3N)	S		N	N	S	N
inet_makeaddr(3N)	S		N	N	S	N
inet_netof(3N)	S		N	N	S	N
inet_network(3N)	S		N	N	S	N
inet_ntoa(3N)	S		N	N	S	N
infinity(3M)	N		N	N	N	N
initgroups(3)	S		S	S	S	N
initscr(3V)	C	The SunOS 4.x version of initscr() is a function while the SunOS release 5.6, SVID, or SVR4 version is a macro that calls initscr32(). If errors occur, the SunOS 4.x initscr() function returns ERR, while the SunOS release 5.6, SVID, or SVR4 version writes an appropriate error message to the standard error and exits.	N	C	C	S
initscr(3V) - SysV	S		N	S	S	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
initstate(3)	S	Now <code>initstate(3C)</code> .	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	Replaced by (int) <code>rint()</code> .	N	N	N	N
isalnum(3V)	S		S	S	S	N
isalpha(3V)	S		S	S	S	N
isascii(3V)	S		S	S	S	N
isatty(3V)	S		S	S	S	N
iscntrl(3V)	S		S	S	S	N
isdigit(3V)	S		S	S	S	N
isendwin(3V) - SysV	S		N	S	S	N
isgraph(3V)	S		S	S	S	N
isinf(3M)	N		N	N	N	N
islower(3V)	S		S	S	S	N
isnan(3M)	S		S	N	N	N

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
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		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	In the SunOS release 4.x software, <code>j0(HUGE)</code> , <code>j1(HUGE)</code> , and <code>jn(4,HUGE)</code> will return zero with no error indication. In the SunOS release 5.6, SVID, or SVR4 software 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.	N	C	C	N
j1(3M)	C		N	C	C	N
jn(3M)	C		N	C	C	N
jrands48(3)	S		N	S	S	N

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	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)	S		N	N	S	N
kvm_write(3K)	S		N	N	N	N
l3tol(3C)	N		N	N	N	N

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
l64a(3)	S		S	S	S	N
label(3X)	S		N	N	N	N
lcong48(3)	S		N	S	S	N
ldaclose(3X)	N		N	N	N	N
ldahread(3X)	N		N	N	N	N
ldaopen(3X)	N		N	N	N	N
ldclose(3X)	N		N	N	N	N
ldexp(3M)	C	The SunOS 4.x version of <code>ldexp()</code> differs from the SunOS release 5.6, ABI, SVID, or SVR4 version only in the case of overflow. The SunOS release 4.x <code>ldexp()</code> returns (+/-) 1.0e999 if the correct value would overflow, while the SunOS release 5.6, ABI, SVID, or SVR4 <code>ldexp()</code> returns (+/-) HUGE (according to the sign of value). Both versions set <code>errno</code> to <code>ERANGE</code> .	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

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
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
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

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
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 software, when log() produces undefined results (for example, log(-1.0)), it returns NaN, with an EDOM error and a DOMAIN matherr. In the SunOS release 5.6, SVID or SVR4 version, it returns -HUGE with an EDOM error and DOMAIN matherr.	N	C	C	N
log10(3M)	C	In the SunOS release 4.x software, when log10() produces undefined results (for example, log10(0)) it returns NaN, with an EDOM error and a DOMAIN matherr. In the SunOS release 5.6, SVID, or SVR4 version, 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 5.6 Status	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 the SunOS release 5.6, SVID, or SVR4 version. <i>termbuf</i> is a pointer to the terminal entry from termcap, which is replaced by <i>terminfo</i> . <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 the SunOS release 5.6, SVID, or SVR4 environment.	N	C	C	S
longname(3V) - SysV	S		N	S	S	N
lrnd48(3)	S		N	S	S	N
lsearch(3)	S		S	S	S	N
lto13(3C)	N		N	N	N	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

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
lwp_destroy(3L)	N		N	N	N	N
lwp_enumerate(3L)	N		N	N	N	N
lwp_errstr(3L)	N		N	N	N	N
lwp_fpset(3L)	N		N	N	N	N
lwp_geterr(3L)	N		N	N	N	N
lwp_getregs(3L)	N		N	N	N	N
lwp_getstate(3L)	N		N	N	N	N
lwp_join(3L)	N		N	N	N	N
lwp_libcset(3L)	N		N	N	N	N
lwp_newstk(3L)	N		N	N	N	N
lwp_perror(3L)	N		N	N	N	N
lwp_ping(3L)	N		N	N	N	N
lwp_resched(3L)	N		N	N	N	N
lwp_resume(3L)	N		N	N	N	N
lwp_self(3L)	N		N	N	N	N
lwp_setpri(3L)	N		N	N	N	N
lwp_setregs(3L)	N		N	N	N	N
lwp_setstkcache(3L)	N		N	N	N	N
lwp_sleep(3L)	N		N	N	N	N

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
lwp_stkcsset(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
mallocmap(3)	S		N	N	N	N
matherr(3M)	S		N	S	S	N
max_normal(3M)	N		N	N	N	N
max_subnormal(3M)	N		N	N	N	N
mblen(3)	S		S	S	S	N
mbstowcs(3)	S		S	S	S	N
mbtowc(3)	S		S	S	S	N
ncmp(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

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
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 <code>template</code> with a letter and the current process ID. The SunOS release 5.6, 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 file name. If the application does not depend on the specific file name (that is, the application only cares that the name is unique), the SunOS 4.x and SunOS release 5.6, ABI, SVID, or SVR4 versions of <code>mktemp()</code> are compatible.	C	C	C	N

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
mlock(3)	S		S	S	S	N
mlockall(3)	S		S	S	S	N
modf(3M)	S		N	S	S	N
mon_break(3L)	N		N	N	N	N
mon_cond_enter(3L)	N		N	N	N	N
mon_create(3L)	N		N	N	N	N
mon_destroy(3L)	N		N	N	N	N
mon_enter(3L)	N		N	N	N	N
mon_enumerate(3L)	N		N	N	N	N
mon_exit(3L)	N		N	N	N	N
mon_waiters(3L)	N		N	N	N	N
moncontrol(3)	A	This routine is replaced by profil(2).	A	A	A	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
monitor(3)	C	<p>The SunOS 4.x monitor() routine differs from the SunOS release 5.6, ABI, SVID, or SVR4 version in the following respects: In the SunOS release 4.x software, to profile the entire program it is sufficient to use:</p> <pre>extern etext; monitor(N_TXTOFF(0), etext, buf, bufsize, nfunc);</pre> <p>While with the SunOS release 5.6, ABI, SVID, or SVR4 monitor() routine, it is sufficient to use:</p> <pre>extern int etext; monitor((int(*)())2, etext, buf, bufsize, nfunc);</pre> <p>In the SunOS release 4.x software, to stop execution monitoring and write the results to the buf defined previously, use:</p> <pre>monitor(0);</pre> <p>While with the SunOS release 5.6, ABI, SVID, or SVR4 monitor() routine, use:</p> <pre>monitor((int(*)())0, (int(*)())0, (WORD*) 0, 0, 0);</pre> <p>The prof(1) command can then be used to examine the results.</p>	C	C	C	N
monstartup(3)	A	This routine is replaced by profil(2).	A	A	A	N
mout(3X)	S		N	N	N	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
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
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 the SunOS release 5.6, 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 the SunOS release 5.6, 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

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
<code>mvaddch(3V)</code> – SysV	S		N	S	S	N
<code>mvaddstr(3V)</code> – SysV	S		N	S	S	N
<code>mvcur(3V)</code>	C	The return value of the SunOS 4.x version of <code>mvcur()</code> is undefined, while the SunOS release 5.6, SVID, or SVR4 version returns <code>OK</code> upon success; otherwise, it returns <code>ERR</code> .	N	C	C	S
<code>mvcur(3V)</code> – SysV	S		N	S	S	N
<code>mvdelch(3V)</code> – SysV	S		N	S	S	N
<code>mvgetch(3V)</code> – SysV	C	In the SunOS release 5.6, SVID, or SVR4 version, 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 software, <code>wrefresh()</code> will not be called under these circumstances.	N	C	C	N
<code>mvgetstr(3V)</code> – SysV	C	See <code>getstr(3V)</code> — Sys V.	N	C	C	N
<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

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
<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 the SunOS release 5.6, SVID, or SVR4 version, 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 software, <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 the SunOS release 5.6, 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

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
mvwprintw(3V) - SysV	S		N	S	S	N
mvwscanw(3V)	C	See wscanw(3V).	N	C	C	S
mvwscanw(3V) - SysV	S		N	S	S	N
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)	A	This routine is replaced by dbm_nextkey(3).	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

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
<code>nlist(3V)</code>	C	The SunOS 4.x version of <code>nlist()</code> returns the number of symbols not found, or -1 on error. The SunOS release 5.6, SVID, or SVR4 version returns 0 on success, and -1 on error. Note that the SunOS release 5.6 <code>nlist()</code> assumes an ELF format file and the 4.1 <code>nlist()</code> works only on a.out format files.	N	C	C	S
<code>nlm_prot(3R)</code>	S		N	N	N	N
<code>nocbreak(3V) - SysV</code>	S		N	S	S	S
<code>nocrmode(3X)</code>	S		N	N	S	N
<code>nodelay(3V) - SysV</code>	S		N	S	S	N
<code>noecho(3V) - SysV</code>	S		N	S	S	S
<code>nonl(3V) - SysV</code>	S		N	S	S	S
<code>nonstandard_arithmetic(3M)</code>		N	N	N	N	N
<code>noraw(3V) - SysV</code>	S		N	S	S	S
<code>notimeout(3V) - SysV</code>	S		N	S	S	N
<code>nrand48(3)</code>	S		N	S	S	N
<code>ntohl(3N)</code>	S		N	N	S	N
<code>ntohs(3N)</code>	S		N	N	S	N
<code>on_exit(3)</code>	A	This routine is replaced by <code>atexit()</code> . Note that functions registered using <code>atexit()</code> are called without arguments.	A	A	A	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
opendir(3V)	C	The SunOS release 5.6, ABI, SVID, or SVR4 DIR structure does not have the <code>dd_bsize</code> and <code>dd_off</code> fields. Also, the SunOS release 5.6, ABI, SVID, or SVR4 <code>dd_loc</code> and <code>dd_size</code> fields are <code>int</code> rather than <code>long</code> . The SunOS release 5.6, ABI, SVID, or SVR4 version includes <code><sys/types.h></code> while the SunOS 4.x version does not. The SunOS release 5.6, 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 5.6 Status	Notes	ABI	SVID	SVR4	BSD
overlay(3V)	C	The SunOS 4.x overlay() is a function while the SunOS release 5.6, SVID, or SVR4 version of overlay(srcwin, dstwin) is a macro that calls _overlap((srcwin), (dstwin), TRUE). The SunOS 4.x version of overlay() return value is undefined, while SunOS release 5.6, 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 overwrite() is a function while the SunOS release 5.6, SVID, or SVR4 version of overwrite(srcwin, dstwin) is a macro that calls _overlap((srcwin), (dstwin), FALSE). The SunOS 4.x overwrite() return value is undefined, while the SunOS release 5.6, 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

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
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 rpcb_getmaps(3N).	A	A	S	N
pmap_getport(3N)	S	This routine is still available, but is superseded by rpcb_getaddr(3N).	A	A	S	N
pmap_rmtcall(3N)	S	This routine is still available, but is superseded by rpcb_rmtcall(3N).	A	A	S	N
pmap_set(3N)	S	This routine is still available, but is superseded by rpcb_set(3N).	A	A	S	N
pmap_unset(3N)	S	This routine is still available, but is superseded by rpcb_unset(3N).	A	A	S	N
pnoutrefresh(3V) - SysV	S		N	S	S	N
pnnp(3R)	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

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
<code>popen(3S)</code>	S		S	S	S	N
<code>pow(3M)</code>	C	In the SunOS release 5.6, SVID, or SVR4 version, the routine returns 0 when $x == 0$ and y is non-positive or when $x < 0$ and y is not integral. For overflow or underflow, <code>pow()</code> returns <code>+/-HUGE</code> 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 the SunOS release 5.6, SVID, or SVR4 version); it returns <code>NaN</code> when $x < 0$ and y not integral, returns <code>+/-infinity</code> when $x == 0$ and $y < 0$. On overflow and underflow, it returns IEEE implementation-dependent values. In the SunOS release 4.x version, since <code>HUGE</code> is defined as <code>+Infinity</code> , <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. In the SunOS release 5.6, SVID, or SVR4 software, these functions set <code>errno</code> to <code>ERANGE</code> .	N	C	C	N
<code>prefresh(3V) - SysV</code>	S		N	S	S	N
<code>printf(3V)</code>	S	See <code>fprintf(3)</code> .	S	S	S	S

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
printw(3V)	C	The SunOS release 5.6, SVID, or SVR4 version of <code>printw()</code> returns the integer <code>ERR</code> upon failure and an integer value other than <code>ERR</code> upon successful completion. The SunOS 4.x version returns <code>void</code> . The SunOS release 5.6, SVID, or SVR4 headers <code><urses.h></code> automatically include 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
printw(3V) - SysV	S		N	S	S	N
prof(3)	A	The <code>profil(2)</code> routine provides similar functionality.	A	A	A	N
psignal(3)	C	The <code>sig</code> argument is defined as an unsigned <code>int</code> in the SunOS 4.x version but is defined as an <code>int</code> in the SVR4 and SunOS release 5.6 versions.	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

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
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)	A	Now random(3C). The drand48(3C) (for SunOS release 5.6, SVID, or SVR4 software) or rand(3C) routines provide 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 5.6 Status	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 the SunOS release 5.6, ABI, SVID or SVR4 software. Applications must migrate to the <code>dirent</code> structure defined in <code><dirent.h></code> . SunOS release 5.6, 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 release 5.6 only have the <code>d_name</code> and <code>d_reclen</code> fields in common. Also, SunOS release 5.6 <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 SunOS 4.x, SVR4, and SunOS release 5.6 <code>dirent</code> structures only have the <code>d_name</code> and <code>d_reclen</code> fields in common. Also, the SunOS release 5.6 <code>dd_loc</code> and <code>dd_size</code> fields are type <code>int</code> rather than type <code>long</code> as in the SunOS 4.x software. The SunOS release 5.6, ABI, SVID, or SVR4 <code>readdir()</code> updates the directory's 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

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
realloc(3)	C	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> . The SunOS release 5.6, ABI, SVID, or SVR4 <code>realloc()</code> does not accept such a pointer.	C	C	C	N
realpath(3)	S		N	N	S	N
re_comp(3)	A	Now <code>re_comp(3C)</code> . 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> .	A	A	A	S
re_exec(3)	A	Now <code>re_exec(3C)</code> . 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> .	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> .	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> . Refer to <code>share(1M)</code> , <code>unshare(1M)</code> , and <code>sharetab(4)</code> man pages for more information.	N	N	N	N

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
remque(3)	S		N	N	S	N
resetterm(3V)	A	This routine is replaced by reset_shell_mode(3).	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
restartterm(3V) - SysV	S		N	S	S	N
rewind(3S)	S		S	S	S	N
rewinddir(3V)	S		S	S	S	N
rex(3R)	S		N	N	N	N
rexec(3N)	S		N	N	S	N
rindex(3)	S	Now rindex(3C).	A	A	A	S
rint(3M)	S		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

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
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)	A	This routine is replaced by def_prog_mode(3X).	N	A	A	N
savetty(3V) - SysV	S		N	S	S	S
scalb(3M)	C	In the SunOS release 5.6, 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 $r == 2$, scalb() is the same as ldexp(3M) routine. On overflow, the routine returns +/- HUGE (depending on the sign of x). On underflow, it returns 0 and sets the errno. In the SunOS 4.x version, the routine computes the value $x * (2^{**n})$ at all times; scalb() is not defined when y is not integral.	N	C	C	N
scalbn(3M)	S		N	N	N	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
scandir(3)	N		N	N	N	S
scanf(3V)	S		S	S	S	N
scanw(3V)	C	In the SunOS release 5.6, SVID, or SVR4 software 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
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
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 the SunOS release 5.6, SVID, or SVR4 version 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

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
scrollok(3V) - SysV	S		N	S	S	S
seconvert(3)	S		N	N	N	N
seed48(3)	S		N	S	S	N
seekdir(3V)	S		S	S	S	N
setac(3)	N		N	N	N	N
setbuf(3V)	S		S	S	S	S
setbuffer(3V)	S		N	N	N	S
set_curterm(3V) - SysV	S		N	S	S	N
setegid(3V)	S		N	N	N	N
seteuid(3V)	S		N	N	N	N
setexportent(3)	A	The /etc/dfs/sharetab file replaces /etc/exports. Refer to share(1M), unshare(1M), and sharetab(4) man pages for more information.	N	N	N	N
setfsent(3)	A	This routine is replaced by fopen(3).	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
sethostent(3N)	S		N	N	S	N
setjmp(3V)	S		S	S	S	S

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
setkey(3)	S		N	S	S	N
setlinebuf(3V)	S		N	N	N	S
setlocale(3V)	C		S	S	S	N
setlogmask(3)	S		N	N	N	N
setmntent(3)	A	The fopen(3) followed by the lockf(3) routines provide similar functionality.	A	A	A	N
setnetent(3N)	S		N	N	S	N
setnetgrent(3N)	S		N	N	N	N
setprotoent(3N)	S		N	N	S	N
setpwaent(3)	N		N	N	N	N
setpwent(3V)	S		S	S	S	N
setpwfile(3V)	N		N	N	N	N
setrgid(3V)	A	This routine is replaced by setgid(2).	A	A	A	N
setrpcent(3N)	S		N	N	S	N
setruid(3V)	A	This routine is replaced by setuid(2).	A	A	A	N
setscrreg(3V) - SysV	S		N	S	S	N
setservent(3N)	S		N	N	S	N
setstate(3)	S	Now setstate(3C).	N	A	A	S
setsyx(3V) - SysV	S		N	S	S	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
set_term(3V) - SysV	S		N	S	S	N
setterm(3V)	C	This is an obsolete call that is replaced by <code>setupterm()</code> in both the SunOS 4.x and SunOS release 5.6 software. See <code>curls_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> man pages for information about SunOS release 5.6 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

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
sigfillset(3V)	S		S	S	S	N
sigfpe(3)	S		N	N	N	N
siginterrupt(3V)	A	The sigaction(2) routine provides similar functionality.	A	A	A	S
sigismember(3V)	S		S	S	S	N
siglongjmp(3V)	S		S	S	S	N
signal(3V)	C	The following SunOS 4.x signal is not defined in the SVR4 and SunOS release 5.6 signal(2) routine: SIGLOST. The following SunOS 4.x signals are not defined in the ABI and SVID signal routine: SIGIO, SIGURG, 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

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
<code>sin(3M)</code>	C	For arguments that are much lower than zero, the SunOS release 5.6, 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 runtime. The error exceptions occur in the IEEE 754 spirit for both versions.	N	C	C	N
<code>sinh(3M)</code>	S		N	S	S	N
<code>single_precision(3M)</code>	N		N	N	N	N
<code>single_to_decimal(3)</code>	S		N	N	N	N
<code>sleep(3V)</code>	S		S	S	S	S
<code>slk_clear(3V)</code> - SysV	S		N	S	S	N
<code>slk_init(3V)</code> - SysV	S		N	S	S	N
<code>slk_label(3V)</code> - SysV	S		N	S	S	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
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	See fprintf(3).	S	S	S	S
sqrt(3M)	C	In the SunOS release 4.x software, when sqrt() produces undefined results (for example, sqrt (-3.0)) it returns NaN, with an EDOM error and a DOMAIN matherr. The SunOS release 5.6, SVID, or SVR4 version returns 0 with an EDOM error and a DOMAIN matherr.	N	C	C	N
srand(3V)	C	In the SunOS release 4.x software, argument seed is defined as int while in the SunOS release 5.6, ABI, SVID, or SVR4 software it is defined as unsigned int.	C	C	C	S
srand48(3)	S		N	S	S	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
srandom(3)	S	Now <code>srandom(3C)</code> . The <code>srand48(3C)</code> (in the SunOS release 5.6, SVID, or SVR4 software) or <code>srand(3C)</code> routines provide similar functionality.	A	A	A	S
sscanf(3V)	S		S	S	S	N
ssignal(3)	S		N	N	S	N
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)	A	This routine is replaced by <code>dbm_store(3)</code> .	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 5.6 Status	Notes	ABI	SVID	SVR4	BSD
strftime(3V)	C	<p>There are some differences in the directives specified in the following formats:</p> <p>%k and %l - Not supported in the SunOS release 5.6 software.</p> <p>%S- the SunOS 4.x software specifies seconds to be in the range of 0-59, while the SunOS release 5.6 software defines seconds to be in the range of 0-61 (allows for leap seconds).</p> <p>%V,%W- Under the SunOS release 4.x software , week number 01 is the first week in January with four or more days in it, while in the SunOS release 5.6 software, week number 01 is the first week in January starting with a Sunday for %U or a Monday for %W.</p> <p>The SunOS 4.1 tm structure contains two fields not present in the SunOS release 5.6 tm structure: tm_zone and tm_gmtoff. Instead, the SunOS release 5.6 version uses the external variable <i>timezone</i> to contain the difference (in seconds) between GMT and local standard time, and the external variable <i>daylight</i> to indicate if daylight savings should be applied.</p> <p>Additionally, the SunOS release 5.6 version uses an external variable <i>tzname</i> to store standard and summer time-zone names. These external variables (<i>timezone</i>, <i>daylight</i>, and <i>tzname</i>) are supported by the SunOS 4.x System 1997 installation option <i>ctime(3V)</i> library routines.</p>	C	C	C	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
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
strpbrk(3)	S		S	S	S	N
strptime(3V)	S		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 strtod() and atof() routines accept <i>inf_form</i> , <i>infinity_form</i> , <i>nan_form</i> , and <i>nanstring_form</i> , while the SunOS release 5.6, ABI, SVID, or SVR4 strtod() and atof() routines do 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 termio(7) interface provides similar functionality.	A	A	A	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
subpad(3V) - SysV	S		N	S	S	N
subwin(3V)	C	The SunOS release 5.6, SVID, or SVR4 routine returns a null pointer if a failure occurs. The SunOS release 5.6, SVID, or SVR4 header file < curses.h > automatically includes the header files <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
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 svc_fdset(3N).	N	S	S	N
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 svc_getrpccaller(3N).	A	A	A	N
svc_getreq(3N)	S	This routine is still available, but is superseded by svc_getreqset(3N).	S	S	S	N
svc_getreqset(3N)	S		S	S	S	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
svc_register(3N)	A	This routine is still available, but it superseded by svc_reg(3N).	A	A	A	N
svc_run(3N)	S		S	S	S	N
svc_sendreply(3N)	S		S	S	S	N
svc_unregister(3N)	A	This routine is still available, but is superseded by svc_unreg(3N).	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
svcfld_create(3N)	A	This routine is still available, but is superseded by svc_fd_create(3N).	A	A	A	S
svccraw_create(3N)	S	This routine is still available, but is superseded by svc_raw_create(3N).	N	N	S	N
svctcp_create(3N)	S	This routine is still available, but is superseded by svc_create(3N), svc_tli_create(3N), and svc_vc_create(3N).	N	N	S	S

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
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.	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> .	N	N	S	S
swab(3)	S		S	S	S	N
sys_siglist(3)	N	Use <code>psignal(3C)</code> .	N	N	N	S
syslog(3)	S		N	N	S	N
system(3)	S		S	S	S	N
t_accept(3N)	S		S	S	S	N
t_alloc(3N)	S		S	S	S	N
t_bind(3N)	S		S	S	S	N
t_close(3N)	S		S	S	S	N
t_connect(3N)	S		S	S	S	N
t_error(3N)	S		S	S	S	N
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

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
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

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
tcflush(3V)	S		S	S	S	N
tcgetattr(3V)	S		S	S	S	N
tcgetpgrp(3V)	S		S	S	S	N
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	The SunOS release 5.6, SVID, or SVR4 software is supporting this routine as a conversion aid and it should not be used in new applications. The SunOS release 5.6, SVID, or SVR4 version returns <code>ERR</code> on failure and an integer value other than <code>ERR</code> upon successful completion.	N	C	C	S
tgetent(3V) - SysV	S		N	S	S	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
tgetflag(3X)	C	The SunOS release 5.6, SVID, or SVR4 software is supporting this routine as a conversion aid and it should not be used in new applications. The SunOS release 5.6, SVID, or SVR4 version returns <code>ERR</code> on failure and an integer value other than <code>ERR</code> upon successful completion.	N	C	C	S
tgetflag(3V) - SysV	S		N	S	S	N
tgetnum(3X)	C	The SunOS release 5.6, SVID, or SVR4 software is supporting this routine as a conversion aid and it should not be used in new applications. The SunOS release 5.6, SVID, or SVR4 version returns <code>ERR</code> on failure and an integer value other than <code>ERR</code> upon successful completion.	N	C	C	S
tgetnum(3V) - SysV	S		N	S	S	N
tgetstr(3X)	C	The SunOS release 5.6, SVID, or SVR4 software is supporting this routine as a conversion aid and it should not be used in new applications. The SunOS release 5.6, SVID, or SVR4 version returns <code>ERR</code> on failure and an integer value other than <code>ERR</code> upon successful completion.	N	C	C	S
tgetstr(3V) - SysV	S		N	S	S	N

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
tgoto(3X)	C	The SunOS release 5.6, SVID, or SVR4 software is supporting this routine as a conversion aid and it should not be used in new applications. The SunOS release 5.6, SVID, or SVR4 version returns <code>ERR</code> on failure and an integer value other than <code>ERR</code> 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 <code>mktime(3C)</code> .	A	A	A	N
timelocal(3V)	S	This routine is the inverse of <code>localtime(3C)</code> .	A	A	A	N
times(3V)	C	The SunOS 4.x <code>times()</code> routine returns time values in units of 1/HZ seconds, where HZ is 60. The SunOS release 5.6, ABI, SVID, or SVR4 <code>times()</code> 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
tmpnam(3S)	S		S	S	S	N

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
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	The SunOS release 5.6, 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	The SunOS release 5.6, 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	The SunOS release 5.6, 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	The SunOS release 5.6, 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

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
tputs(3V)	C	The SunOS release 5.6, SVID, or SVR4 software supports this routine as a conversion aid. It should not be used in new applications. The SunOS release 5.6, 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
traceoff(3V) - SysV	S		N	N	S	N
traceon(3V) - SysV	S		N	N	S	N
tsearch(3)	S		S	S	S	N
ttynname(3V)	S		S	S	S	N
ttyslot(3V)	S		N	N	S	N
twalk(3)	S		S	S	S	N
typeahead(3V) - SysV	S		N	S	S	N
tzset(3V)	C	See ctime(3V).	C	C	C	N
tzsetwall(3V)	A	This routine is replaced by tzset(3C).	A	A	A	N
ualarm(3)	S	Now ualarm(3C). The setitimer(2) system call with the <i>which</i> argument set to ITIMER_REAL provides similar functionality.	N	A	A	S

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
ulimit(3C)	S	The SVR4 and SunOS release 5.6 <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's file descriptor table).	C	C	S	N
unctrl(3V) – SysV	S		N	S	S	S
ungetc(3S)	S	The SVR4 and SunOS release 5.6 <code>ungetc()</code> guarantees to push back four characters, so it is compatible with the SunOS 4.x <code>ungetc()</code> . In the SunOS release 4.x software, <code>ungetc()</code> is guaranteed to push back 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
ungetch(3V) – SysV	S		N	S	S	N
user2netname(3N)	S		S	S	S	N
usleep(3)	S	Now <code>usleep(3C)</code> . The <code>setitimer(2)</code> or <code>select(3C)</code> routines provide similar functionality.	N	A	A	S

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
utime(3V)	C	The SunOS 4.x <code>utime()</code> and SunOS release 5.6, ABI, SVID, or SVR4 <code>utime()</code> differ in the type of the second argument. In the SunOS release 4.x software, argument <code>timep</code> points to an array of two <code>time_t</code> values, while in the SunOS release 5.6, ABI, SVID, or SVR4 version, argument <code>times</code> points to the <code>utimbuf</code> structure (which contains two <code>time_t</code> members).	C	C	C	N
valloc(3)	S		N	N	S	N
varargs(3)	S		N	N	N	N
vfprintf(3V)	C	See <code>vprintf(3V)</code> .	C	C	C	S
vidattr(3V) - SysV	S		N	S	S	N
vidputs(3V) - SysV	S		N	S	S	N
vlimit(3C)	A	This routine is replaced by <code>getrlimit(2)</code> .	A	A	A	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
vprintf(3V)	C	The SunOS 4.x vprintf(), vfprintf(), and vsprintf() routines are incompatible with the SunOS release 5.6, ABI, SVID, or SVR4 version of these routines because of variable format list differences. In the SunOS release 4.x software, (<i>va_alist</i> (defined in <varargs.h>) is used in a function header to declare a variable argument list (for example, void function(<i>va_alist</i>)). In the SunOS release 5.6, ABI, SVID, or SVR4 version 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).	N	N	N	N
vtimes(3C)	A	This routine is replaced by getrusage(2).	N	N	N	N
vwprintw(3V) - SysV	S		N	S	S	N
vwscanw(3V) - SysV	S		N	S	S	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
waddch(3V)	C	The characters in the SunOS release 5.6 software are chtype (long) if CHTYPE is not defined differently for < curses.h>. The SunOS release 5.6, SVID, or SVR4 version returns ERR on failure and an integer value other than ERR upon successful completion. The SunOS release 5.6, 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
waddch(3V) - SysV	S		N	S	S	N
waddstr(3V)	C	The SunOS release 5.6, SVID, or SVR4 version of waddstr(3V) returns ERR (-1) on failure. The SunOS release 5.6, 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

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
wattrset(3V) - SysV	S		N	S	S	N
wclear(3V)	C	The SunOS release 5.6, SVID, or SVR4 version of <code>wclear()</code> routine always returns (<code>OK = 0</code>) upon success while the SunOS 4.x software returns <code>void</code> . The SunOS release 5.6, 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
wclear(3V) - SysV	S		N	S	S	N
wclrtoobot(3V)	C	The SunOS release 5.6, SVID, or SVR4 version of <code>wclrtoobot(3V)</code> routine always returns (<code>OK = 0</code>) upon success while the SunOS 4.x software returns <code>void</code> . The SunOS release 5.6, 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
wclrtoobot(3V) - SysV	S		N	S	S	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
wclrtoeol(3V)	C	The SunOS release 5.6, SVID, or SVR4 routine always returns (OK = 0) upon success while the SunOS 4.x software returns void. The SunOS release 5.6, 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	The size of wchar_t is short in the SunOS 4.x software and long in the SunOS release 5.6 software.	S	S	S	N
wctomb(3)	S	The size of wchar_t is short in the SunOS 4.x software and long in the SunOS release 5.6 software.	S	S	S	N
wdelch(3V)	C	In the SunOS release 5.6, SVID, or SVR4 software this routine may be a macro, while it always is in the SunOS 4.x software. The SunOS release 5.6, 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

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
wdeleteln(3V)	C	In the SunOS release 5.6, SVID, or SVR4 software this routine may be a macro, while it always is in the SunOS 4.x software. The SunOS release 5.6, SVID, or SVR4 version returns <code>ERR</code> on failure and an integer value other than <code>ERR</code> upon successful completion.	N	C	C	S
wdeleteln(3V) - SysV	S		N	S	S	N
wechochar(3V) - SysV	S		N	S	S	N
werase(3V)	C	In the SunOS release 5.6, SVID, or SVR4 software this routine returns <code>OK(0)</code> or a non-negative integer if <i>immedok</i> is set. The SunOS release 5.6, 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	The SunOS release 5.6, SVID, or SVR4 version of <code>wgetch()</code> returns <code>ERR</code> on failure and an integer value other than <code>ERR</code> upon successful completion. The SunOS release 5.6, SVID, or SVR4 version also has additional support for function keys.	N	C	C	S

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
wgetch(3V) – SysV	C	In the SunOS release 5.6, SVID, or SVR4 software, 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 software, wrefresh() will not be called under these circumstances.	N	C	C	N
wgetstr(3V)	C	The SunOS release 5.6, SVID, or SVR4 version of wgetstr() returns ERR on failure and an integer value other than ERR upon successful completion.	N	C	C	S
wgetstr(3V) – SysV	C	See getch(3V) — Sys V.	N	C	C	N
winch(3V) – SysV	S		N	S	S	S
winsch(3V)	C	The SunOS release 5.6, SVID, or SVR4 version of winsch() 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
winsertln(3V)	C	The SunOS release 5.6, 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 5.6, or the SVID or SVR4.	N	C	C	S

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
winsertln(3V) - SysV	S		N	S	S	N
wmove(3V)	C	The SunOS release 5.6, SVID, or SVR4 version of <code>wmove()</code> returns <code>ERR</code> on failure and an integer value other than <code>ERR</code> 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	The SunOS release 5.6, SVID, or SVR4 version of <code>wprintw()</code> returns <code>ERR</code> on failure and an integer value other than <code>ERR</code> upon successful completion. The SunOS 4.x version returns <code>void</code> . SunOS release 5.6, or the SVID or SVR4 header <code><curses.h></code> automatically includes the headers <code><stdio.h></code> and <code><unctrl.h></code> and if <code>CURS_PERFORMANCE</code> is defined it defines most commonly used routines as macros for increased performance.	N	C	C	S
wprintw(3V) - SysV	S		N	S	S	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
wrefresh(3V)	C	The SunOS release 5.6, SVID, or SVR4 version of wrefresh() returns (ERR = -1) on failure and some other integer on success while SunOS 4.x returns void. SunOS release 5.6, 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
wrefresh(3V) - SysV	S		N	S	S	N
wscanw(3V)	C	The SunOS release 5.6, SVID, or SVR4 version of wscanw() returns an int containing the number of fields mapped by the call while the SunOS 4.x version returns void. The SunOS release 5.6, 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
wscanw(3V) - SysV	S		N	S	S	N
wsetscrreg(3V) - SysV	S		N	S	S	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
wstandend(3V)	C	This is a curses(3V) function that clears all window attributes using attrset(0). The SunOS 4.x version always returns undefined while the SunOS release 5.6, SVID, or SVR4 standout() routine always returns 1 (success).	N	C	C	S
wstandend(3V) - SysV	S		N	S	S	N
wstandout(3V)	C	This is a curses(3V) function that sets the A_STANDOUT attribute to enable the terminals best standout mode for a window. The SunOS 4.x version uses attron(A_STANDOUT) for this function and returns undefined. The SunOS release 5.6, SVID, or SVR4 standout() routine is the same as: attron(A_STANDOUT) 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 xdr_authsys_parms(3N).	A	A	A	N

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
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_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
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

TABLE C-1 Library Routines Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
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 5.6 Status	Notes	ABI	SVID	SVR4	BSD
y0(3M)	C	In the SunOS release 4.x software, when these routines have undefined results they return NaN, with an EDOM error and a DOMAIN matherr. In the SunOS release 5.6, SVID, or SVR4 software, they return -HUGE with an EDOM error and a DOMAIN matherr. In the SunOS release 4.x software, y0(HUGE), y1(HUGE), yn(9,HUGE) will return zero with no error indication.	N	C	C	N

TABLE C-1 Library Routines Reference Table (continued)

SunOS release 4.x	SunOS release 5.6 Status	Notes	ABI	SVID	SVR4	BSD
y1(3M)	C	In the SunOS release 4.x software, when these routines have undefined results they return NaN, with an EDOM error and a DOMAIN matherr. In the SunOS release 5.6, SVID, or SVR4 software, they return -HUGE with an EDOM error and a DOMAIN matherr. In the SunOS release 4.x software, y0(HUGE), y1(HUGE), yn(9,HUGE) will return zero with no error indication.	N	C	C	N
yn(3M)	C	In the SunOS release 4.x software, when these routines have undefined results they return NaN, with an EDOM error and a DOMAIN matherr. In the SunOS release 5.6, SVID, or SVR4 software, they return -HUGE with an EDOM error and a DOMAIN matherr. In the SunOS release 4.x software, y0(HUGE), y1(HUGE), yn(9,HUGE) 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 2.6 environment.

Using the Reference Table

- If an interface is listed as “changed” (C), a brief description of differences between the SunOS 4.x and Solaris 2.6 file is provided.
- If an interface is listed as “the same” (S), the Solaris 2.6 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 2.6 interfaces, see the man Pages(4): File Formats.

System Files

TABLE D-1 File Formats Reference Table

SunOS release 4.x	SunOS release 5.6Status	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	Security audit trail file
audit_control(5)	N	Control information for system audit daemon
audit_data(5)	N	Current information on audit daemon
auto.home(5)	C	Automount map for home directories
auto.master(5)	C	Automount map for home directories
bar(5)	N	Tape archive file format
boards.pc(5)	N	ATN and XTN compatible boards for DOS windows
bootparams(5)	S	Boot parameter database
cpio(5)	S	Format of <code>cpio</code> archive
crontab(5)	S	Table of times to run periodic jobs
dir(5)	A	Format of directories
dump(5)	C	Incremental dump format
environ(5V)	C	User environment
ethers(5)	S	Ethernet address to <i>hostname</i> database or NIS domain
exports(5)	A	Directories to export to NFS clients
fbtab(5)	C	Frame buffer table

TABLE D-1 File Formats Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6Status	Notes
fcntl(5)	C	File control options
fs(5)	C	Format of a 4.2 (ufs) file system volume
fspec(5)	S	Format specification in text files
fstab(5)	A	Static file system mounting table, mounted file systems table
ftpusers(5)	S	List of users prohibited by FTP
gettytab(5)	N	Terminal configuration database
group(5)	S	Group file
group.adjunct(5)	N	Group security data file
holidays(5)	C	Prime/non-prime table for System V accounting
hosts(5)	S	Host-name database
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
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

TABLE D-1 File Formats Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6Status	Notes
netgroup(5)	S	List of network groups
netmasks(5)	S	Network mask database
netrc(5)	S	File for ftp remote login data
networks(5)	S	Network name database
passwd(5)	C	Password file
passwd.adjunct(5)	N	User security data file. See shadow(4).
phones(5)	S	Remote-host phone number database
plot(5)	N	Graphics interface
printcap(5)	A	Printer capability database
proto(5)	S	Prototype job file for at
protocols(5)	S	Protocol name data base
publickey(5)	S	Public key database
queuedefs(5)	S	Queue description file for at, batch, and cron
rasterfile(5)	S	Sun's file format for raster images
remote(5)	S	Remote host description file
resolv.conf(5)	S	Configuration file for domain-name system resolver
rfmaster(5)	N	
rgb(5)	N	Available colors (by name) for coloredit
rhosts(5)	S	Trusted hosts by system and by user
rmtab(5)	S	Remote mounted file-system table

TABLE D-1 File Formats Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6Status	Notes
rootmenu(5)	A	Root menu specification for SunView
rpc(5)	S	RPC program number database
sccsfile(5)	S	Format of an SCCS history file
services(5)	S	Internet services and aliases
sm(5)	S	<code>in.statd</code> directory and file structures
statmon(5)	S	<code>statd</code> 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
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 database
terminfo(5V)	S	Terminal capability database
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

TABLE D-1 File Formats Reference Table *(continued)*

SunOS release 4.x	SunOS release 5.6Status	Notes
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 database
xtab(5)	N	Directories to export to NFS clients
ypaliases(5)	N	NIS aliases for sendmail
ypfiles(5)	S	NIS database and directory structure
ypgroup(5)	N	NIS group file
yppasswd(5)	N	NIS password file
ypprintcap(5)	N	NIS printer capability database

/ and /usr File Systems Changes

This Appendix shows the layout of directories in the / and /usr file systems. Appendix A, explains differences in commands within these directories.

Layout of the / File System

Table E-1 shows the layout of the SunOS release 5.6 / 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/rdisk	Raw disk devices
/dev/pts	Pseudo terminal slave devices
/dev/rmt	Raw tape devices
/dev/sad	Entry points for the STREAMS Administrative Driver

TABLE E-1 Directories in the / File System *(continued)*

Directory	Description
/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
/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

TABLE E-1 Directories in the / File System *(continued)*

Directory	Description
/etc/saf	Service Access Facility (SAF) files, including FIFOs
/etc/skel	Default profile scripts for new user accounts
/etc/sm	Status monitor information
/etc/sm.bak	Backup copy of status monitor information
/etc/tm	Trademark files; contents displayed at boot time
/etc/uucp	Configuration information for uucp
/export	Default root of the exported file-system tree
/home	Default root of a subtree for user directories
/kernel	Subtree of loadable kernel modules, including the base kernel itself as /kernel/unix
/mnt	Temporary mount point for file systems
/opt	Root of a subtree for add-on application packages
/opt/SUNWspro	Mount/installation point for unbundled language products
/platform	Subtree of loadable kernel modules
/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

TABLE E-1 Directories in the / File System *(continued)*

Directory	Description
<code>/var/lp</code>	Line printer subsystem logging information
<code>/var/mail</code>	Directory where users' mail is kept
<code>/var/news</code>	Community service messages (not to be confused with USENET-style news)
<code>/var/nis</code>	NIS+ databases
<code>/var/opt</code>	Root of a subtree for various files associated with optional software packages
<code>/var/options</code>	Provides package compatibility with pre-SunOS 5.0 packages
<code>/var/preserve</code>	Backup files for <code>vi</code> and <code>ex</code> editors
<code>/var/sadm</code>	Databases maintained by the software package management utilities
<code>/var/saf</code>	System Access Facility (SAF) logging and accounting files
<code>/var/spool</code>	Directories for spooled temporary files
<code>/var/spool/cron</code>	Spool files for <code>cron</code> and <code>at</code>
<code>/var/spool/locks</code>	Spooling lock files
<code>/var/spool/lp</code>	Line printer spool files
<code>/var/spool/mqueue</code>	Mail queued for delivery
<code>/var/spool/pkg</code>	Spooled packages
<code>/var/spool/uucp</code>	Queued <code>uucp</code> jobs
<code>/var/spool/uucppublic</code>	Files deposited by <code>uucp</code>
<code>/var/tmp</code>	Directory for temporary files not cleared during boot sequence

TABLE E-1 Directories in the / File System *(continued)*

Directory	Description
/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, and the like)
/usr/kernel	Additional modules

TABLE E-2 Directories in the `/usr` File System *(continued)*

Directory	Description
<code>/usr/kvm</code>	Implementation architecture-specific binaries and libraries
<code>/usr/lib</code>	Various program libraries, architecture-dependent databases, and binaries not invoked directly by the user
<code>/usr/lib/acct</code>	Accounting scripts and binaries
<code>/usr/lib/dict</code>	Database files for the <code>spell</code> command
<code>/usr/lib/class</code>	Scheduling class-specific directories containing executables for <code>priocntl</code> and <code>dispadm</code> commands
<code>/usr/lib/font</code>	Font description files for <code>troff</code>
<code>/usr/lib/fs</code>	File system type dependent modules; not invoked directly by the user
<code>/usr/lib/iconv</code>	Conversion tables for <code>iconv</code>
<code>/usr/lib/libp</code>	Profiled libraries
<code>/usr/lib/locale</code>	Internationalization and localization databases
<code>/usr/lib/localedef</code>	Locale source file for <code>localedef</code> .
<code>/usr/lib/lp</code>	Line printer subsystem databases and back-end executables
<code>/usr/lib/mail</code>	Auxiliary programs for the mail subsystem
<code>/usr/lib/netsvc</code>	Internet network services
<code>/usr/lib/nfs</code>	Auxiliary NFS-related programs and daemons
<code>/usr/lib/pics</code>	PIC archives needed to build the runtime linker
<code>/usr/lib/refer</code>	Preprocessor for <code>nroff/troff</code>
<code>/usr/lib/sa</code>	Scripts and commands for the system activity report package
<code>/usr/lib/saf</code>	Auxiliary programs and daemons related to the Service Access Facility (SAF)

TABLE E-2 Directories in the `/usr` File System *(continued)*

Directory	Description
<code>/usr/lib/spell</code>	Auxiliary <code>spell</code> -related programs and databases
<code>/usr/lib/uucp</code>	Auxiliary <code>uucp</code> -related programs and daemons
<code>/usr/local</code>	Commands local to a site
<code>/usr/net/servers</code>	Entry points for foreign name-service requests related by the listener
<code>/usr/oasys</code>	Files pertaining to the optional FACE package
<code>/usr/old</code>	Programs that are being phased out
<code>/usr/openwin</code>	Mount or installation point for OpenWindows software
<code>/usr/sadm</code>	Various files and directories related to system administration
<code>/usr/sadm/bin</code>	Binaries for use by FMLI scripts
<code>/usr/sadm/install</code>	Executables and scripts for package management
<code>/usr/sbin</code>	Executables for system administration
<code>/usr/sbin/static</code>	Statically linked versions of selected programs from <code>/usr/bin</code> and <code>/usr/sbin</code> ; used to recover from broken dynamic linking
<code>/usr/share</code>	Architecture-independent databases
<code>/usr/share/lib</code>	Architecture-independent databases
<code>/usr/share/lib/keytables</code>	Keyboard layout description tables
<code>/usr/share/lib/mailx</code>	Help files for <code>mailx</code>
<code>/usr/share/lib/nroff</code>	Terminal tables for <code>nroff</code>
<code>/usr/share/lib/pub</code>	Various data files
<code>/usr/share/lib/spell</code>	Auxiliary <code>spell</code> -related databases and scripts
<code>/usr/share/lib/tabset</code>	Tab-setting escape sequences

TABLE E-2 Directories in the /usr File System *(continued)*

Directory	Description
/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.

Summary Tables

TABLE F-1 Basic Commands

SunOS Release 4.x	Solaris 2.6	Comments
<code>lpr</code>	<code>lp</code>	Basic default print command
<code>lpr -P printer</code>	<code>lp -d printer</code>	Specifying a printer with the print command
<code>lpq</code>	<code>lpstat -o</code>	Check the print queue of the default printer
<code>lpq -P printer</code>	<code>lpstat -o printer</code>	Check the status of a specific printer and list print IDs
	<code>lpstat -a</code>	Determine which printers are available (in the SunOS release 4.x software, you would check the <code>/etc/printcap</code> file)
<code>lprm print job#</code>	<code>cancel request ID</code>	Cancel a print job
	<code>cancel printer</code>	Alternate method for canceling a currently active print job

TABLE F-1 Basic Commands *(continued)*

SunOS Release 4.x	Solaris 2.6	Comments
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

SunOS Release 4.x	Solaris 2.6	Comments
dump	ufsdump	For backing up file systems or specified files
exportfs	share <i>resources</i>	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 <i>resource</i>	Used to make resources unavailable
mount -a	mountall	Mount all file systems specified in /etc/vfstab, where the mountall option is set
restore	ufsrestore	For restoring files dumped to backup media
showmount -d	dfmounts <i>option</i>	Lists mounted NFS file systems where option specifies machine name
showmount -e	dfshares <i>option</i>	Lists shared (exported) NFS file systems
umount -a	umountall	Unmount all file systems in /etc/vfstab, other than root, /proc, /var, and /usr

TABLE F-3 Files and Directories

SunOS Release 4.x	Solaris 2.6	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/mntab	/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

SunOS Release 4.x	Solaris 2.6	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

TABLE F-4 Daemons and Standard Processes *(continued)*

SunOS Release 4.x	Solaris 2.6	Comments
/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

TABLE F-5 File and Command Differences

SunOS Release 4.x	Solaris 2.6
ac	sar
add_services	pkgadd
arch	uname -m
bar	Use cpio -H bar to retrieve
biff -n	chmod -o-x /dev/tty
biff -y	chmod -o+x /dev/tty
cc	Not available
dbxtool	debugger
df	df -k
dket1	Not available
dkinfo	prvtoc
du	du -k

TABLE F-5 File and Command Differences *(continued)*

SunOS Release 4.x	Solaris 2.6
dump	ufsdump
dumpfs	Not available
etherfind	snoop
exportfs	share
extract_files	Not available
extract_patch	Not available
extract_unbundled	pkgadd
fastboot	reboot or init -6
fasthalt	init -0
hostid	sysdef -h
hostname	uname -n
intr	Not available
leave	Use cron and at
lint	Not available
load	pkgadd
loadc	pkgadd
load_package	Not available
lpc	lpadmin
lpd	lpsched

TABLE F-5 File and Command Differences *(continued)*

SunOS Release 4.x	Solaris 2.6
lpq	lpstat
lpr	lp
lprm	cancel
lpptest	Not available
mach	uname -p
modstat	mount -a
mount	mount -F <i>fstype</i> [<i>options</i>]
mountall	modinfo
mount_tfs	mount -F <i>fstype</i>
pax	cpio
paxcpio	cpio
portmap	rpcbind
printenv	env
ps -a	ps -e
ps -aux	ps -el
pstat	sar
pstat -s	swap -s
rdump	ufsdump
restore	ufsrestore

TABLE F-5 File and Command Differences *(continued)*

SunOS Release 4.x	Solaris 2.6
rm_client	admintool
rm_services	Not available
rpc.etherd	Not available
rpc.lockd	lockd
rpc.mountd	mountd
rpc.read	Not available
rpc.rquotad	Not available
rpc.showfhd	showfhd
rpc.statd	statd
rpc.user_agentd	Not available
rpc.yppasswdd	Not available
rpc.yppupdated	ypupdated
rrestore	ufsrestore
rusage	Not available
startup	Not available
swapon	swap -a
sys-config	admintool
umountall	umount -a
umount-tfs	umount -F <i>fstype</i>

TABLE F-5 File and Command Differences *(continued)*

SunOS Release 4.x	Solaris 2.6
unload	pkgrm
update	fsflush
uptime	who -b
users	who -q
vipw	Not available
wall	Not available
whereis	Not available
whoami	id
ybatchupd	Not available
yppasswd	Use nispasswd for NIS+
ypserv	Not available

Glossary

Architecture	The specific components of a computer system and the way they interact with one another. From a Solaris 2.6 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 enables existing SunOS release 4.x applications, both statically and dynamically linked, to run under SunOS release 5.6 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 <code>/usr</code> and <code>/usr/kvm</code> 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 <i>disk slice</i> .

Disk slice	A discrete portion of a disk, configured during installation. Slices were referred to as partitions under the SunOS 4.1.x and System V Release 3 software.
Diskless	A system whose <code>root</code> , <code>swap</code> , and <code>/usr</code> 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 2.6 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 2.6 distribution. This can be on either 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 2.6 kernel. Two systems have the same kernel architecture if the same Solaris 2.6 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 2.6 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 5.6 software is grouped and distributed in packages. Packages are also the standard way to deliver unbundled Sun and third-party software.
SAC	Services Access Facility. A SunOS release 5.6 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 2.6 software.
Standalone	A system that does not depend on a server for its <code>root</code> , <code>swap</code> , or <code>/usr</code> 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 5.6 software distribution: for example, the SunPro [™] compilers.

Index

\$
%

/ file system, 29, 76, 387, 391
/4lib directory, 391
/5lib directory, 87
, xiii

A

a.out system file, 89, 382
a641 library routine, 290
abort library routine, 290
abs library routine, 290
ac command, 202
accept system call, 258
access system call, 258
accounting files
 scripts and binaries, 392
 Service Access Facility (SAF), 390
 system, 389
 UUCP, 121
accounting system, xiii
 configuration information
 directory, 388
user accounts, xiii
acct command, 240
/acct directory, 388, 392
acct system call, 258
acct system file, 382
acctcms command, 202
acctcom command, 202
acctcon1 command, 202
acctcon2 command, 202

acctdisk command, 202
acctdusg command, 202, 214
acctmerg command, 202
accton command, 202
acctprc1 command, 202
acctprc2 command, 203
acctwtmp command, 203
acos library routine, 290
acosh library routine, 291
adb command, xiii, 156, 157, 203
adbgen command, 203
add_client command, 203
add_drv command, 68, 186, 187, 198
add_services command, 203
addbib command, 203
addch library routine, 291
addexportent library routine, 291
adding
 devices, 61, 186, 187, 198
 network devices, 23
 NIS+ information, 22
 port monitor services, 114
 port monitors, 114
 software packages, 12, 170
addmntent library routine, 291
addstr library routine, 291
adjacentscreens command, 203
adjtime system call, 258
/adm directory, 389
admin command, 172, 203
/.Admin directory, 121
administrators, , *see* system
 administrators\x0d

- administration of users and groups, 49
- Administration Tool, xiii
 - files directory, 87, 394
- User Account Manager, xiii
- admintool
 - described, 8, 20, 21
 - for user and group administration, 49
 - Serial Port Manager, 113
- admintool command, xiii, 20, 21
 - add_client command vs., 203
 - files directory, 394
 - rm_client command vs., 238
- adv command, 203
- aedplot command, 203
- aging passwords, 40, 41
- agt_create library routine, 291
- agt_enumerate library routine, 291
- agt_trap library routine, 291
- aint library routine, 291
- aiocancel library routine, 291
- aioread library routine, 291
- aiowait library routine, 291
- aiowrite library routine, 291
- alarm library routine, 291
- aliasadm command, 20
- aliases system file, 382, 397
- align_equals command, 203
- alloca library routine, 291
- alphasort library routine, 291
- analyze command, 203
- anint library routine, 292
- annuity library routine, 292
- ANSI C compiler, 139, 148
- application concurrency, 9, 195
- applications
 - determining compatibility of, 168
 - porting, reasons for, 35
- apropos command, 204
- ar command, 172, 204, 236
- ar system file, 382
- arc library routine, 292
- arch command, 204
- architecture-independent databases,
 - directories for, 393
- architecture-specific information,
 - directories for, 392

- archives, converting to random
 - libraries, 173
- arp command, 204
- as command, 172, 204
- asctime library routine, 292
- ASET (automated security enhancement
 - tool), 8, 42
- Asian language character sets, 179
- asin library routine, 292
- asinh library routine, 292
- assert library routine, 292
- async_daemon system call, 258
- at command, 204, 225, 390
- AT&T systems
 - device naming, 15
 - SVR4 features excluded from Solaris
 - operating
 - environment, 9, 10
 - sysadm menu utility, 10
- atan library routine, 292
- atan2 library routine, 292
- atanh library routine, 293
- atexit library routine, 335
- atof library routine, 293
- atoi library routine, 293
- atol library routine, 293
- atoplot command, 204
- atq command, 205
- atrm command, 205
- attroff library routine, 293
- attron library routine, 293
- attrset library routine, 293
- audioio ioctl, 161
- audit command, 205
- audit system call, 258
- audit.log system file, 382
- audit_args library routine, 293
- audit_control system file, 382
- audit_data system file, 382
- audit_text library routine, 293
- audit_warn command, 205
- auditd command, 205
- auditon system call, 258
- auditsvc system call, 258
- auth_destroy library routine, 293

authdes_create library routine, 293
 authdes_getucrd library routine, 293
 authdes_seccreate library routine, 293
 authnone_create library routine, 293
 authsys_create_default library routine, 293
 authsys_seccreate library routine, 293
 authunix_create library routine, 293
 authunix_create_default library routine, 293
 auto configuration, 8
 auto.home system file, 206, 382
 auto.master system file, 206, 382
 auto_home system file, 206
 auto_master system file, 206
 autofs program, 18, 90, 91
 automated security enhancement tool (ASET), 8, 42
 automatic loading, device drivers, 9
 automount command, 206
 automounting, 18, 90–92, 206
 autopush command, 197
 awk command, 206

B

backing up, xiii
 changes in, 96
 installation and, 27, 30
 ufsdump command, xiii
 Backup CoPilot, 27, 96
 backup files, directory for vi and ex editors, 390
 banner command, 206
 bar command, 96, 206
 bar system file, 382
 basename command, 206
 batch command, 206
 baudrate library routine, 293
 bc command, 206
 bcmp library routine, 294
 bcopy library routine, 294
 beep library routine, 294
 Berkeley compatibility package, directories for, 394
 BFS (boot file system), 79
 bgplot command, 206
 biff command, 206

/bin directory, , *see* /usr/bin directory\x0d
 bin-mail command, 207
 binaries, directories for, 388, 391, 393
 Binary Compatibility Package, 35, 36, 168, 391
 bind system call, 258
 bindresvport library routine, 294
 bindtextdomain library routine, 180
 biod command, 207, 398
 block disk devices, directory for, 82, 188, 387
 block I/O daemon, 398
 boards.pc system file, 382
 boot block, 67, 185
 boot command
 changes in, 66, 67, 185, 207
 device additions and, 61, 66
 reconfiguration boot, 186
 boot file system (BFS), 79
 boot.sun4c.sunos.4.1 command, 67, 186
 bootblk command, 186
 booting, xiii, 65, 69, 185
 changes in, 65, 67, 68, 185
 from PROM, 67
 bootparam library routine, 294
 bootparamd command, 207
 bootparams database, 382
 bootsd command, 67
 Bourne shell, 45–47, 242
 restricted, 41, 240
 box library routine, 294
 brk system call, 258
 BSD Source Compatibility Package, 35, 36
 bsearch library routine, 294
 byteorder library routine, 295
 bzero library routine, 295

C

C compilation system, directory for, 87, 391
 C compiler, 139, 148, 174
 C program tools, 174
 C shell, 45–47, 211
 C2conv command, 202

C2unconv command, 202
CACHEFS (cache file system), 74, 78
cal command, 207
calendar command, 207
Calendar Manager, 4, 7
calloc library routine, 295
callrpc library routine, 295
cancel command, 23, 111, 207, 227, 395
capitalize command, 207
captoinfo command, 207
cat command, 207
catclose library routine, 295
catgetmsg library routine, 181, 295
catgets library routine, 180, 295
catman command, 207
catopen library routine, 295
cb command, 174, 207
cbc_crypt library routine, 295
cbreak library routine, 295, 299
cbrt library routine, 295
cc command, 174, 207
/ccs directory, , *see* /usr/ccs
 directory\x0d
cd command, 207
CD-ROM devices
 improving performance, 74, 78
 install4x program and, 103, 104
 managing, 24, 61, 63
 naming conventions, 59
 Solaris 2.6 requirement for, 26
CD-ROM file system (HSFS), 62, 77, 92
cdc command, 172, 208
/cdrom directory, 62, 92
ceil library routine, 295
cfgetispeed library routine, 295
cfgetospeed library routine, 295
cflow command, 174, 208
cfree library routine, 296
cfsetispeed library routine, 296
cfsetospeed library routine, 296
changing
 NIS+ information, 22
 passwords, 40, 41
 shells, 45, 46
 system run levels, 69, 70
character sets, 179
chargefee command, 208
chdir system call, 258
check command, 111
CHECK library routine, 289
checkeq command, 208
checking, xiii
 file systems, 80, 95
 port monitor status, 114
 printer status, 110, 395
 software package installation, 13, 170
checknr command, 208
chfn command, 208
chgrp command, 208
chkey command, 208
chmod command, 206, 208
chmod system call, 259
chown command, 209
chown system call , 259
chroot library routine, 180, 209
chroot system call , 259
chrtbl command, 209
chsh command, 209
circle library routine, 296
ckpacct command, 209
/class directory, 392
clear command, 209
clear library routine, 296
clear_colormap command, 209
clear_function command, 209
clearerr library routine, 296
clearing inodes, 79
clearok library routine, 296
click command, 209
clients, xiii
 diskless/dataless, 101, 108
NIS (Network Information Services),
 clients, xiii
clnt_broadcast library routine, 296
clnt_call library routine, 296
clnt_control library routine, 296
clnt_create library routine, 296, 297
clnt_destroy library routine, 296
clnt_dg_create library routine, 297
clnt_freeres library routine, 296
clnt_geterr library routine, 296

- clnt_pcreateerror library routine, 296
- clnt_perrno library routine, 296
- clnt_perror library routine, 296
- clnt_raw_create library routine, 297
- clnt_spcreateerror library routine, 297
- clnt_sperrno library routine, 297
- clnt_sperror library routine, 297
- clnt_tli_create library routine, 297
- clnt_vc_create library routine, 297
- clntraw_create library routine, 297
- clnttcp_create library routine, 297
- clntudp_bufcreate library routine, 297
- clntudp_create library routine, 297
- clock command, 209
- clock library routine, 297
- close system call, 259
- closedir library routine, 297
- closelog library routine, 297
- closepl library routine, 298
- clri command, 79, 209
- clrtoob library routine, 298
- clrtoeol library routine, 298
- clusters, 12, 26
- cmdtool command, 209
- cmp command, 209
- COFF, 174, 175
- col command, 209
- colcrt command, 210
- colldef command, 210
- colltbl command, 210
- coloredit command, 210
- colrm command, 210
- comb command, 172, 210
- comm command, 210
- command log file (UUCP), 121
- Command Tool (OpenWindows), 209
- command-line interface
 - for user and group administration, 49
- command-line utilities, for software
 - package administration, 13
- commands, xiii
 - changes in, xiii, 35
 - device driver, 198
 - file system, 79, 82, 90
 - local, 393
 - messed, 180
 - NFS, 118
 - printer, 111, 112
 - quick reference, 395, 396
 - Service Access Facility (SAF), 24, 53, 114
 - table of, 199, 254
 - Volume Management, 63
- Commands reference table, 199, 254
- comment section, object files, 174
- Common Desktop Environment (CDE)
 - development history, 129
 - Session Manager
 - Window Manager, 130
- community service messages, directory
 - for, 390
- compatibility
 - cross-functional hardware, 4, 6
 - determining for applications, 168
 - SunOS release 4.x with release 5.6, 35, 48
- compatibility packages, 35
- compilers, 139, 148, 174
- compound library routine, 298
- compress command, 210
- concurrency, 9, 195
- config command, 68, 184, 186, 210
- Config file, 121
- configuration, xiii
 - auto, 8
 - kernel, 17
 - new features, 25, 26
- configuration files, , *see* /etc
 - directory\x0d
- system configuration, xiii
- connect system call, 259
- consistency checking, , *see* checking\x0d
- cont library routine, 298
- conventions, notation, xvii, xviii
- conversion tables, for iconv command, 392
- convert4x program, 102, 108
- converting
 - archives to random libraies, 173
 - files, 33, 34
- copyright file, 169

- copysign library routine, 298
- copywin library routine, 298
- Core System Support, 12
- cos library routine, 298
- cosh library routine, 299
- cp command, 210
- cpio command, 98
 - bar command vs., 206
 - changes in, 99–101, 210
 - described, 98, 101
 - pax command vs., 234
 - paxcpio command vs., 234
 - support for, 96
- cpio system file, 382
- cpp command, 172, 210
- crash command, 210
- /crash directory, 389
- creat system call, 260
- creating
 - file systems, 80, 94
 - software packages, 170
- crmode library routine, 299
- cron command, 210, 225, 389, 390
- /cron directory, 389, 390
- /cron.d directory, 388
- crontab command, 210
- crontab system file, 382
- cross-functional compatibility, 4
- crtplot command, 211
- crypt command, 211
- crypt library routine, 289, 299
- _crypt library routine, 289
- cscope command, 174
- cs command, 45–47, 211
- .cshrc file, 47
- csplit command, 211
- ctags command, 211
- ctermid library routine, 299
- ctime library routine, 300
- ctrace command, 174, 211
- cu command, 211
- curs_set library routine, 301
- cuserid library routine, 301
- customizing
 - kernel, 68
 - man command search path, 53
 - man command search path, 54
 - port monitors, 114
 - user environments, 47, 48
- customizing with Style Manager, 134
- cut and paste, 7
- cut command, 211
- cv_broadcast library routine, 301
- cv_create library routine, 301
- cv_destroy library routine, 301
- cv_enumerate library routine, 301
- cv_notify library routine, 301
- cv_send library routine, 301
- cv_wait library routine, 301
- cv_waiters library routine, 301
- cxref command, 174, 211
- cylinder group maps, 14
- cylinder groups, 14

D

- daemons, quick reference, 397, 398
- data access, common, 74
- Data Link Provider Interface (DLPI), 178
- data transfer
 - restoring, 31, 35
 - saving, 26
- data types, expanded, 9
- databases
 - architecture-independent, 393
 - bootparams, 382
 - gettytab, 383
 - hosts, 383
 - inetd.conf, 383
 - locale, 383, 392
 - netmasks, 384
 - networks, 384
 - NIS+, 390
 - phones, 384
 - printcap, 22, 34, 74, 110, 111, 384, 397
 - printer subsystem, 392
 - protocols, 384
 - publickey, 384
 - rpc, 385
 - software package management utility
 - maintained, 88, 390
 - spell command, 392, 393

- termcap, 74, 385
- terminfo, 74, 110, 111, 385, 394, 397
- vgrindefs, 386
- whatis, 54, 204
- windex, 54, 204
- yp, 386, 391
- dataless clients, , *see* diskless/dataless clients\x0d
- date command, 180, 211
- dbconfig command, 212
- dbm_clearerr library routine, 301
- dbm_close library routine, 301
- dbm_delete library routine, 301
- dbm_error library routine, 301
- dbm_fetch library routine, 301, 308
- dbm_firstkey library routine, 301, 309
- dbm_nextkey library routine, 301, 334
- dbm_open library routine, 301
- dbm_store library routine, 302, 353
- dbmclose library routine, 302
- dbminit library routine, 302
- dbx command, 156, 174, 212
- dbxtool command, xiii, 156, 174, 212
- dc command, 212
- dcheck command, 212
- dd command, xiii, 96, 97, 212
- DDI (Device Driver Interface), 9, 192, 197
- DDI-compliant device drivers,
 - defined, 193
- DDI/DKI (Device Driver Interface/Driver Kernel Interface), 9, 192, 193, 198
- DDI/DKI compliant
 - defined, 193
- ddi_create_minor_node command, 187
- Debugger (Sun WorkShop), 156
- debuggers, xiii, 80, 156, 174
- truss command, xiii
- decimal_to_double library routine, 302
- decimal_to_extended library routine, 302
- decimal_to_floating library routine, 302
- decimal_to_single library routine, 302
- def_prog_mode library routine, 302, 345
- def_shell_mode library routine, 302
- /default directory, , *see* /etc/default directory\x0d
- defaults
 - automounting, 76
 - directories, 75, 76
 - directory for, 40, 84, 388
 - file systems, 75, 76, 81
 - root access, 40
 - shell, 45, 46
 - swap device, 74, 78
 - Volume Management file system, 75
 - window system, 48
- defaults_from_input command, 212
- defaults_merge command, 212
- defaults_to_indentpro command, 212
- defaults_to_mailrc command, 212
- defaultsed command, 212
- del_curterm library routine, 302
- delay_output library routine, 302
- delch library routine, 302
- delete library routine, 302
- deleteln library routine, 302
- deleting, , *see* removing\x0d
- delta command, 172, 213
- delwin library routine, 302
- /demo directory, 391
- deroff command, 213
- des command, 213
- des_crypt library routine, 302
- des_setparity library routine, 302
- DeskSet, 4, 7
- desktop
 - overview, 130
- desktop integration services, 7
- /dev directory
 - changes in, 15, 57, 68, 82, 387
 - described, 188, 387
 - /devices directory and, 187
 - reconfiguration boot and, 186
- /dev/dsk directory, 82, 188, 387
- /dev/ksyms file, 157
- /dev/pts directory, 83, 387
- /dev/rdisk directory, 82, 188, 387
- /dev/rmt directory, 83, 387
- /dev/sad directory, 83, 387
- /dev/sd1g command, 68
- /dev/term directory, 83, 388

- dev_info nodes, 197
- Developer System Support, 12
- developers, 137, 198
 - Binary Compatibility Package and, 37, 168
 - compilers, 139, 148
 - debuggers, 80, 156
 - device configuration, 186, 189
 - device drivers, 191, 198
 - internationalization, 178, 181
 - linkers, 148, 155
 - networking features, 177, 178
 - Solaris features for, 8, 9
 - STREAMS, 196, 198
 - system configuration, 183, 186
 - tools and resources, 159, 174
 - transition information overview, 137
- device administration, 57, 63
 - adding devices, 61, 186, 187, 198
 - automatic loading, 9
 - improving performance, 74, 78
 - information reporting, 7, 59, 61, 193
 - loading devices, 9, 18, 184, 198
 - naming conventions, 15, 57, 59, 187, 189
 - removing devices, 186, 198
 - unloading devices, 18, 184, 198
 - Volume Management for, 24, 61, 63, 75
- device configuration, xiii, 34, 84, 186, 189
- Device Driver Interface (DDI), 9, 192, 197
- Device Driver Interface/Driver Kernel Interface (DDI/DKI), 9, 192, 193, 198
- device drivers, xiii, 191, 198
 - changes in, 191
 - commands, 198
 - DDI-compliant, 193
 - directories, 89
 - interface compatibility, 9
 - interface types, 9, 192, 193
 - leaf drivers, 197, 198
 - nexus drivers, 197
 - porting considerations, 195, 196
 - STREAMS, 196, 198
- specific devices, xiii
- device naming, xiii
 - CD-ROMS, 59
 - changes in, 15, 57, 59
 - developer's perspective, 187, 189
 - disks, 58, 59, 188
 - instance names, 189
 - logical names, 188
 - physical names, 188
- device special file system (SPECFS), 78
- device tree, 187, 197, 198
- /devices directory, 187
- devinfo command, 59, 61, 193, 213
- devinfo tree, 197
- devnm command, 213
- df command, 59, 79, 93, 213
- dfmounts command, 92, 396
- DFS (distributed file system)
 - administration, 118
 - /dfs directory, , see /etc/dfs directory\x0d
- dfshares command, 92, 396
- dgettext library routine, 180
- /dict directory, 392
- diff command, 214
- diff3 command, 214
- diffmk command, 214
- dir system file, 382
- dircmp command, 214
- directories, xiii
 - changes in, 73, 82, 85
 - default, 75, 76
 - kernel module, 18, 184
 - monitoring, 92
 - names, 73
 - path name generation for file systems, 80
 - quick reference, 397, 398
 - remote, automounting, 18
 - searching at link time, 152, 153
- dirname command, 214
- dis command, 174, 214
- disabling port monitor services, 114
- discover4x program, 102
- disk devices
 - directories for, 82
 - information reporting, 59, 61, 79, 93

- names of disks attached to system, 27
- naming conventions, 58, 59, 188
- partition information, saving, 27
- disk slices, 13
- diskette devices, managing, 24, 61, 63
- diskette file system, automounting of, 62, 92
- diskless/dataless clients, Solaris 2.6 server to support SunOS release 4.x, 101
- diskusg command, 214
- dispadmin command, 392
- distributed file system (DFS)
 - administration, 118
- dkctl command, 214
- DKI (Driver Kernel Interface), 9, 192
- dkinfo command, 27, 59, 60, 214
- dkio ioctl, 159, 160
- dlclose library routine, 302
- dlerror library routine, 302
- dlopen library routine, 303
- DLPI (Data Link Provider Interface), 178
- dlsym library routine, 303
- dmesg command, 214
- dn_comp library routine, 303
- dn_expand library routine, 303
- dname command, 214
- DNS (domain name system)
 - described, 123, 124
 - NIS+ (Network Information Services Plus) vs., 124, 125
- document tools, using, 51
- dodisk command, 215
- dollar sign (library routine)
- domain name system, , *see* DNS (domain name system)\x0d
- domainname command, 215
- dorfs command, 215
- dos2unix command, 215
- double_to_decimal library routine, 303
- doupdate library routine, 303
- drag and drop, 7
- draino library routine, 303
- drand48 library routine, 303, 319, 341, 348
- Driver Kernel Interface (DKI), xiii, 9, 192
- drivers, , *see* device drivers\x0d

- /drv directory, 89
- drvconfig program, 187
- /dsk directory, 82, 188, 387
- du command, 59, 60, 93, 215
- dumbplot command, 215
- dump command, xiii
 - changes to old, 96, 215
 - link checks using, 154, 155
 - new, 174
 - new 218, xiii
 - quick reference, 396
- ufsdump command, xiii
- dump system file, 382
- dumpfs command, 215
- dumpkeys command, 215
- dup system call, 261
- dup2 system call, 261
- dynamic kernel, xiii
 - defined, 8, 17, 183, 184
- dynamic linking, 9
- dysize library routine, 303

E

- e command, 216
- e-mail, , *see* mail\x0d
- ecb_crypt library routine, 303
- echo command, 216
- echo library routine, 303
- echochar library routine, 303
- econvert library routine, 303
- ecvt library routine, 303
- ed command, 216
- edata library routine, 303
- edit command, 216
- edquota command, 216
- EEPROM command, 216
- EFT (Extended Fundamental Types), 27
- egrep command, 216
- eject command, 216
- ELF (executable and linking format) files
 - compiler and, 139
 - kernel modules to run, 89
 - linker and, 148
- enabling port monitor services, 114
- encrypt library routine, 303

- end library routine, 303
- End User System Support, 12
- endac library routine, 303
- endexportent library routine, 304
- endfsent library routine, 304
- endgraent library routine, 304
- endgrent library routine, 304
- endhostent library routine, 304
- endmntent library routine, 304
- endnetent library routine, 304
- endnetgrent library routine, 304
- endprotoent library routine, 304
- endpwaent library routine, 304
- endrpwent library routine, 304
- endrpcent library routine, 304
- endservent library routine, 304
- endttyent library routine, 304
- endusershell library routine, 304
- endwin library routine, 305
- enroll command, 216
- Entire Distribution, 12
- env command, 216, 234
- environ system file, 382
- environment initialization files
 - customizing user environments, 47, 48
- environments, *see* Solaris operating environment
- eqn command, 216
- erand48 library routine, 305
- erase library routine, 305
- eraseschar library routine, 305
- erf library routine, 305
- erfc library routine, 305
- errno library routine, 305
- errno values, 255
- error command, 172, 216
- error messages
 - file system commands, 80
 - Volume Management, 63
- /etc directory, xiii
 - changes in, 74, 81, 83, 85, 388, 389
 - described, 76, 83, 388
 - /etc/.login file, 47
 - /etc/acct directory, 388
 - /etc/aliases file, 382, 397
 - /etc/config command, 68, 184, 186, 210
 - /etc/cron.d directory, 388
 - /etc/default directory, 40, 84, 388
 - /etc/default/fs file, 81
 - /etc/default/login file, 40
 - /etc/default/passwd file, 40
 - /etc/default/su file, 40
 - /etc/dfs directory, 388
 - /etc/dfs/dfstab file, 34, 94
 - /etc/dfs/fstype file, 81
 - /etc/dfs/sharetab file, 291, 397
 - /etc/exports file, 34, 93, 291, 382, 397
 - /etc/fs directory, 82, 388
 - /etc/fstab file
 - described, 27, 383, 397
 - /etc/vfstab vs., 33, 34, 84, 397
 - /etc/group file, 34, 383
 - /etc/inet directory, 388
 - /etc/inet directory, 84
 - /etc/init.d scripts, 236, 388
 - /etc/inittab file, 67, 68
 - /etc/lib directory, 388
 - /etc/lp directory, 23, 74, 84, 110, 388
 - /etc/lp directory, 110
 - /etc/lp/printers directory, 111
 - /etc/mail directory, 388
 - /etc/mail/aliases file, 397
 - /etc/mail/sendmail.cf file, 34
 - /etc/mnttab file, 397
 - /etc/mntab file, 383, 397
 - /etc/net directory, 388
 - /etc/netgroup file, 34, 384
 - /etc/opt directory, 84, 89, 388
 - /etc/passwd file, 34, 40, 384, 397
 - /etc/printcap database, replacement
 - of, 22, 34, 74, 110, 111, 384, 397
 - /etc/profile file, 47
 - /etc/rc scripts, 68, 69, 83, 186, 236, 388, 398
 - /etc/rc.boot script, 68, 69, 83, 186, 236
 - /etc/rc.local script, 68, 69, 83, 186, 236, 398
 - /etc/rc.single script, 68, 83, 186
 - /etc/rc scripts, xiii
 - /etc/rc scripts, 68, 83, 186

- /etc/rc.d scripts, 83, 186, 388, 398
- /etc/rcS script, 68, 84, 186
- /etc/rcS.d script, 84, 388, 398
- /etc/rmmount.conf file, 63
- /etc/saf directory, 84, 389
- /etc/sendmail.cf file, 34
- /etc/shadow file, 34, 40, 86, 397
- /etc/skel directory, 47, 389
- /etc/sm directory, 389
- /etc/sm.bak directory, 389
- /etc/system file
 - described, 61, 68, 184, 186
 - moddir variable, 18, 184
- /etc/tm directory, 389
- /etc/ttytab file, 34, 385
- /etc/uucp directory, 34, 119, 389
- /etc/uucp/Config file, 120, 121
- /etc/uucp/Grades file, 119
- /etc/uucp/Limits file, 119, 121
- /etc/vfstab file
 - described, 84, 397
 - fields, 84, 86
 - merging /etc/fstab file into, 33, 34
 - specifying file systems in, 92
- /etc/vold.conf file, 63
- /etc/xtab file, 386, 397
- extext library routine, 305
- ether library routine, 305
- ether_aton library routine, 305
- ether_hostton library routine, 305
- ether_line library routine, 305
- ether_ntoa library routine, 305
- ether_ntohost library routine, 305
- etherd command, 216
- etherfind command, 216
- Ethernet drivers, 178
- ethers system file, 382
- EUC (extended UNIX code), 179
- ex command, 216
- ex editor, backup files directory, 390
- exc_bound library routine, 305
- exc_handle library routine, 305
- exc_notify library routine, 306
- exc_on_exit library routine, 306
- exc_raise library routine, 306
- exc_unhandle library routine, 306

- exc_uniqpatt library routine, 306
- /exec directory, 89
- execl library routine, 306
- execle library routine, 306
- execlp library routine, 306
- executable and linking format files, , *see*
 - ELF (executable and linking format) files\x0d
- executables, building, 151
- execv library routine, 306
- execve system call, 261
- execvp library routine, 306
- exit library routine, 307
- _exit system call, 258
- exp library routine, 307
- exp10 library routine, 307
- exp2 library routine, 307
- expand command, 216
- expm1 library routine, 308
- /export directory, 101, 389
- /export/home directory, 206
- exported file systems, , *see* shared file
 - systems\x0d
- exportent library routine, 308
- exportfs command, 91, 94, 217, 396
- exports file, 34, 93, 291, 382, 397
- expr command, 217
- exstr command, 174
- Extended Fundamental Types (EFT), 27
- extended UNIX code (EUC), 179
- extended_to_decimal library routine, 308
- extract_files command, 217
- extract_patch command, 217
- extract_unbundled command, 217

F

- fabs library routine, 308
- FACE package, directories for, 393, 394
- false command, 217
- fastboot command, 66, 72, 217
- fasthalt command, 68, 70, 72, 217
- fbtab system file, 382
- fchdir system call, 261
- fchmod system call, 261
- fchown system call, 261

- fchroot system call, 261
- fclose library routine, 304, 308
- fcntl system call, 262
- fcntl system call, 274
- fcntl system file, 383
- fconvert library routine, 308
- fcvt library routine, 308
- fdformat command, 217
- FDFS (file descriptor file system), 74, 78
- fdopen library routine, 308
- feof library routine, 308
- ferror library routine, 308
- fetch library routine, 308
- ff command, 79
- fflush library routine, 308
- ffs library routine, 308
- fgetc library routine, 308
- fgetgraent library routine, 308
- fgetgrent library routine, 308
- fgetpwaent library routine, 309
- fgetpwent library routine, 309
- fgets library routine, 309
- fgrep command, 217
- FIFOFS (FIFO/pipe file system), 74, 78
- file command, 217
- file descriptor file system (FDFS), 74, 78
- File Manager, 4, 24, 62
- file system administration, *see* file systems\x0d
- file systems, xiii, 73
 - added, 75
 - automounting, 18, 90–92, 206
 - backing up, 27, 30, 96
 - CD-ROM devices with, 62
 - changes in, 73, 75
 - format, 27
 - layout, 387, 394
 - location, 73
 - names, 32, 73
 - checking, 80, 95
 - commands
 - changes in, 90
 - described, 79, 82
 - locations, 82
 - syntax, 80, 81
 - using, 90
 - creating, 80, 94
 - cylinder groups and, 14
 - debugger, 80
 - default, 75, 76, 81
 - disk slices and partitions, 13
 - diskette devices with, 62
 - error message, 80
 - image copy of, 80
 - kernel modules for implementing, 89
 - labels, 80, 96, 224
 - listing systems to save, 29
 - monitoring, 92
 - mounting, 18, 34, 80, 90, 92, 230, 271, 396
 - automounting, 18, 90–92, 206
 - mount table, 397
 - path name list generation, 80
 - pseudo, 74, 78
 - remote
 - automounting, 18
 - default type, 81
 - mounting, 80, 90
 - unmounting, 80
 - restoring, 97
 - saving information, 27
 - specifying in /etc/vfstab file, 92
 - supported types, 77, 79
 - System V, 9, 79
 - type determination, 80
 - unmounting, 80, 91
 - unsupported SVR4, 79
 - utility overview, 7
 - virtual architecture (VFS), 77, 82
- , xiii
- file systems, *see* shared file systems\x0d
 - exported,
 - shared,
- file systems, *see* temporary file systems\x0d
 - temporary,
- file_to_decimal library routine, 309
- fileno library routine, 309
- files, xiii
 - backing up, 27, 30, 96
 - converting, 33, 34

- environment initialization, 47, 48
- information reporting, 79
- listing files to save, 29
- merging, 32, 33
- monitoring, 92
- names, 32, 73, 79
- opening using file descriptors, 74
- pipe, pseudo file system for, 74
- quick reference, 397, 398
- restoring, 97
- searching for, 93
- system, 381, 386
- , xiii
- filio ioctls, 159, 161
- filter library routine, 309
- filters, image, unsupported, 23
- find command, 93, 217
- finger command, 218
- fingerd command, 218
- finite library routine, 309
- firstkey library routine, 309
- fixterm library routine, 309
- flash library routine, 309
- floatingpoint library routine, 309
- flock system call , 262
- floor library routine, 309
- /floppy directory, 62, 92
- flushinp library routine, 309
- flusok library routine, 309
- FMLI scripts, binaries directory for, 393
- fmod library routine, 310
- fmt command, 218
- fmt_mail command, 218
- fold command, 218
- /font directory, 392
- fontedit command, 218
- fonts
 - description files directory for troff, 392
 - device-independent, 51
- fopen library routine, 310, 347, 348
- foption command, 218
- foreign name service requests, entry
 - points for, 393
- fork system call , 262
- format command, 27, 218
- fp_class library routine, 310
- fpa_download command, 218
- fparel command, 218
- fpathconf system call, 262
- fpaversion command, 218
- fprintf library routine, 310
- fpurel command, 218
- fputc library routine, 310
- fputs library routine, 310
- fpuversion command, 218
- fread library routine, 310
- free library routine, 296, 310
- freopen library routine, 310
- frexp library routine, 310
- from command, 218
- /fs directory, 82, 83, 89, 388, 392
- fs system file, 383
- fscanf library routine, 310
- fsck command, 80, 95, 212, 218
- fsck_cdrom command, 218
- fsdb command, 80, 221
- fseek library routine, 310
- fsflush command, 250
- fsirand command, 219
- fspec system file, 383
- /fstab system file, , *see* /etc/fstab
 - file\x0d
- fstat system call, 262
- fstatfs system call, 262
- fstatvfs system call, 262
- fstyp command, 80, 215
- fsync system call , 262
- ftell library routine, 310
- ftime library routine, 310
- ftok library routine, 310
- ftp command, 219
- ftpd command, 219
- ftpusers system file, 383
- ftruncate system call, 263
- ftw library routine, 310
- fumount command, 219
- func_to_decimal library routine, 311
- fusage command, 219
- fuser command, 219
- fwrite library routine, 311
- fwtmp command, 219

G

/games directory, 391
games, binaries and data directory, 391
gamma library routine, 311
garbageclines library routine, 311
gcd library routine, 311
gconvert library routine, 311
gcore command, 219
gcvt library routine, 311
generic file system commands, 79, 82
generic_args command, 219
get command, 172, 219
get_alarm command, 219
get_myaddress library routine, 314
get_selection command, 219
getacdir library routine, 311
getacflg library routine, 311
getacinfo library routine, 311
getacmin library routine, 311
getauditflagsbin library routine, 311
getauditflagschar library routine, 311
getauid system call, 263
getbegyx library routine, 311
getc library routine, 311
getcap library routine, 311
getch library routine, 312
getchar library routine, 312
getcwd library routine, 180, 312, 316
getdate library routine, 355
getdents system call, 263
getdiretries system call, 263
getdomainname system call, 263
getdtablesize system call, 263
getegid system call, 263
getenv library routine, 312
geteuid system call, 263
getexportent library routine, 312
getexportopt library routine, 312
getfauditflags library routine, 313
getfsent library routine, 313
getfsfile library routine, 313
getfsspec library routine, 313
getfstype library routine, 313
getgid system call, 264
getgraent library routine, 313
getgranam library routine, 313
getgrent library routine, 313
getgrgid library routine, 313
getgrnam library routine, 313
getgroups system call, 264
gethostbyaddr library routine, 313
gethostbyname library routine, 313
gethostent library routine, 313
gethostid system call, 264
gethostname system call, 264
getitimer system call, 264
getlogin library routine, 313
getmaxyx library routine, 313
getmntent library routine, 314
getmsg system call, 264
getnetbyaddr library routine, 314
getnetbyname library routine, 314
getnetent library routine, 314
getnetgrent library routine, 314
getnetname library routine, 314
getopt command, 219
getopt library routine, 314
getoptcv command, 219
getopts command, 219
getpagesize system call, 264
getpass library routine, 314
getpeername system call, 265
getpgid system call, 265
getpgrp system call, 265
getpid system call, 265
getppid system call, 266
getpriority system call, 266
getprotobyname library routine, 314
getprotobynumber library routine, 314
getprotoent library routine, 315
getpublickey library routine, 315
getpw library routine, 315
getpwaent library routine, 315
getpwanam library routine, 315
getpwent library routine, 315
getpwnam library routine, 315
getpwuid library routine, 315
getrlimit system call, 263, 266, 278, 366
getrpcbyname library routine, 315
getrpcbynumber library routine, 315
getrpcent library routine, 315

getrpcport library routine, 315
 getrusage system call, 266
 getrusage system call, 367
 gets library routine, 315
 getsecretkey library routine, 315
 getservbyname library routine, 315
 getservbyport library routine, 315
 getservent library routine, 315
 getsockname system call, 266
 getsockopt system call, 266
 getstr library routine, 315
 getsubopt library routine, 315
 getsyx library routine, 316
 gettable command, 219
 gettext library routine, 181, 316
 gettimeofday system call, 266
 gettmode library routine, 316
 getttyent library routine, 316
 getttynam library routine, 316
 getty command, 220
 gettytab database, 383
 getuid system call, 267
 getusershell library routine, 316
 getut library routine, 181
 getvfsany library routine, 313
 getvfssent library routine, 313
 getvfsfile library routine, 313
 getw library routine, 316
 getwd library routine, 316
 getyx library routine, 316
 gftool command, 220
 gigipolot command, 220
 glob command, 220
 glossary, 403, 405
 gmtime library routine, 316
 goto command, 220
 gpconfig command, 220
 gprof command, 174, 220
 graph command, 220
 graphical user interfaces (GUIs), xiii

- admintool, 20, 21
- for installation, 4, 8
- Software Manager, 12, 13
- XView Windows Toolkit, 171

graphics libraries, 7
 grep command, 220

group system file, 34, 383
 group.adjunct system file, 383
 groups

- administering, 50
- groups command, 220
- groups, administering, 8, 49
- groupware productivity tools, 4
- grpauth library routine, 316
- grpck command, 220
- gsignal library routine, 317
- gty library routine, 317
- GUIs, , *see* graphical user interfaces (GUIs)\x0d
- gxtest command, 220

H

halfdelay library routine, 317
 halt command, 66, 70, 72, 220
 halting, , *see* shutting down\x0d
 has_ic library routine, 317
 has_il library routine, 317
 hashcheck command, 220
 hashmake command, 220
 hashstat command, 220
 hasmntopt library routine, 317
 hcreate library routine, 317
 hdestroy library routine, 317
 head command, 220
 header files, directory for, 196
 help command, 172, 221
 Help files, for mailx command, 393
 help_open command, 221
 holidays system file, 383
 /home directory, 389
 /home file system, 76
 \$HOME/.login file, 47
 \$HOME/.profile file, 47
 \$HOME/.cshrc file, 47
 \$HOME/ file, 47
 Host Manager, xiii
 host2netname library routine, 317
 hostid command, 221
 hostname command, 221
 hosttrfs command, 221
 hosts database, 383

hosts.equiv system file, 383
hp7221plot command, 221
hpplot command, 221
hsearch library routine, 317
HSFS (CD-ROM file system), 62, 77, 92
htable command, 221
HUGE library routine, 289
HUGE_VAL library routine, 290
hypot library routine, 317

I

i386 command, 221
iAPX286 command, 221
icheck command, 221
Icon Edit tool (OpenWindows), 221
iconedit command, 221
iconv command, conversion tables, 392
/iconv directory, 392
id command, 221, 252
ID data types, expanded, 9
idload command, 221
idlok library routine, 317
ieee_flags library routine, 317
ieee_functions library routine, 317
ieee_handler library routine, 318
ieee_retrospective library routine, 318
ifconfig command, 221
ilogb library routine, 318
image copy, file systems, 80
image filters, unsupported, 23
Image Tool, 4, 7
imemtest command, 221
implot command, 221
in.comsat command, 221
in.fingerd command, 222
in.ftpd command, 222
in.named command, 222, 230
in.rexcd command, 237
in.rexd command, 237
in.rexecd command, 222
in.rlogind command, 222
in.routed command, 222
in.rshd command, 222
in.rwhod command, 222
in.talkd command, 222
in.telnetd command, 222

in.tftpd command, 222
in.tnamed command, 222
in.uucpd command, 222
inch library routine, 318
/include directory, 87, 391
include header files, directory for, 391
indent command, 175, 222
indent.pro system file, 383
indentpro_to_defaults command, 222
index library routine, 318
indxbib command, 222
/inet directory, 84, 388
inet_lnaof library routine, 318
inet_makeaddr library routine, 318
inet_netof library routine, 318
inet_network library routine, 318
inet_ntoa library routine, 318
inetboot command, 67, 185, 186
inetd command, 222
inetd.conf database, 383
infinity library routine, 318
infocmp command, 223
information reporting, xiii
 device administration, 7, 59, 61, 193
 disk devices, 59, 61, 79, 93
 file systems, 79, 95, 96
 kernel modules, 183
 software packages, 13, 170
init command
 changes in, 68, 223
 commands replaced by, 66, 217
 described, 67
 using, 68, 70
init.d scripts, 236, 388
initgroups library routine, 318
initialization files, , *see* environment
 initialization files\x0d
initialization scripts, , *see* scripts\x0d
initialization states, changing, 68, 70
initscr library routine, 318
initstate library routine, 319
inittab file, 67, 68
inline command, 175, 223
innetgr library routine, 319
inodes, clearing, 79

- input_from_defaults command, 223
- insch library routine, 319
- insert_brackets command, 223
- insertln library routine, 319
- insque library routine, 319
- install command, 223
- install4x program, 102, 103
- installation, 25, 33, 34
 - install phase, 25, 31
 - new features, 25, 26
 - overview, 8, 25
 - post-installation phase, 25, 31, 34
 - pre-installation phase, 26
 - backing up file systems, 27, 30
 - listing files and file systems to save, 29
 - listing system components to save, 28
 - network installation order, 30
 - overview, 25
 - saving disk partition information, 27
 - saving file system information, 27
 - saving metadvice configuration information, 28
 - software packages, 12, 13, 392
- installboot command, 67, 185, 223
- installtxt command, 180, 223
- instance device names, 189
- integrity checking, , *see* checking\x0d
- internat system file, 383
- internationalization, 178, 181, 392
- Internet services
 - configuration information
 - directory, 84, 388
 - network services directory, 392
 - security, 39
- interoperability, 5
- interprocess communication utilities, 7
- intr command, 223
- intrflush library routine, 319
- ioctl requests, 159, 161, 267
 - STREAMS, 196, 197
- iostat command, 223
- ipalloc library routine, 319
- ipallocald command, 224

- ipcrm command, 224
- ipcs command, 224
- irint library routine, 319
- isalnum library routine, 319
- isalpha library routine, 319
- isascii library routine, 319
- isatty library routine, 319
- iscntrl library routine, 319
- isdigit library routine, 319
- isendwin library routine, 319
- isgraph library routine, 319
- isinf library routine, 319
- islower library routine, 319
- isnan library routine, 319
- isnormal library routine, 320
- isprint library routine, 320
- ispunct library routine, 320
- issecure library routine, 320
- isspace library routine, 320
- issubnormal library routine, 320
- isupper library routine, 320
- isxdigit library routine, 320
- iszero library routine, 320
- itom library routine, 320

J

- j0 library routine, 320
- j1 library routine, 320
- jn library routine, 320
- job grading, 119, 121
- join command, 224
- rand48 library routine, 320

K

- kadb command, xiii, 156, 157, 224
- Kerberos security, 43
- kernel, 16, 75
 - architecture-specific, 26
 - booting and, 67
 - configuration, 17
 - crash dump directory, 389
 - customizing, 68
 - debugging a live, 157
 - dynamic, 8, 17, 183

- layout, 18, 184
- modules, 16, 75
 - directory search path, 18, 184, 389, 391
 - information reporting, 183
 - loading, 18, 184
 - location, 16, 18, 26
 - unloading, 18, 184
- MT (multithreaded), 9, 195
 - name of, 16
- /kernel directory, 18, 75, 88, 184, 389, 391
- /kernel, 16, 75
- /kernel/drv directory, 89
- /kernel/exec directory, 89
- /kernel/fs directory, 89
- /kernel/misc directory, 89
- /kernel/sched directory, 89
- /kernel/strmod directory, 89
- /kernel/sys directory, 89
- /kernel/unix directory, 26, 67, 89, 184
- key_decryptsession library routine, 321
- key_encryptsession library routine, 321
- key_gendes library routine, 321
- key_setsecret library routine, 321
- keyboard layout description tables, 393
- keyenvoy command, 224
- keylogin command, 224
- keylogout command, 224
- keyname library routine, 321
- keypad library routine, 224
- keyserv command, 224
- /keytables directory, 393
- keytables system file, 383
- kgmon command, 224
- kill command, 224
- kill system call, 267
- killchar library routine, 321
- killpg system call, 267
- klm_prot library routine, 321
- Korn shell, 45–47
 - restricted, 41
- ksh command, 45–47
- ksyms file, 157
- /kvm directory, 392
- kvm_close library routine, 321
- kvm_getcmd library routine, 321

- kvm_getproc library routine, 321
- kvm_getu library routine, 321
- kvm_nextproc library routine, 321
- kvm_nlist library routine, 321
- kvm_open library routine, 321
- kvm_read library routine, 321
- kvm_setproc library routine, 321
- kvm_write library routine, 321

L

- l3tol library routine, 181, 321
- l64a library routine, 322
- label library routine, 322
- labelit command, 80, 96, 224
- langinfo library routine, 181
- language products, mount/installation
 - point for, 389
- large organizations, advantages of Solaris
 - for, 6
- last command, 224
- lastcomm command, 224
- lastlogin command, 224
- lcong48 library routine, 322
- ld command, 172, 225
- ldaclose library routine, 322
- ldahread library routine, 322
- ldaopen library routine, 322
- ldclose library routine, 322
- ldconfig command, 225
- ldd command, 225
- ldexp library routine, 322
- ldfcn library routine, 322
- ldfhread library routine, 322
- ldgetname library routine, 322
- ldlinit library routine, 322
- ldlitem library routine, 322
- ldlread library routine, 322
- ldlseek library routine, 323
- ldnlseek library routine, 323
- ldnrseek library routine, 323
- ldnshread library routine, 323
- ldnsseek library routine, 323
- ldohseek library routine, 323
- ldopen library routine, 323
- ldrseek library routine, 323

- ldshread library routine, 323
- ldsseek library routine, 323
- ldtbindex library routine, 323
- ldtbread library routine, 323
- ldtbseek library routine, 323
- leaf drivers, 197, 198
- leaf nodes, 197
- leave command, 225
- leaveok library routine, 323
- lex command, 172, 225
- lfind library routine, 323
- lgamma library routine, 323
- /usr/share/lib directory, xiii
- /lib directory, , *see* /etc/lib directory
- libc directory, 180
- libintl directory, 180
- /libp directory, 166, 392
- libraries
 - changes in, 163, 167
 - converting archives to random, 173
 - dynamic linking of, 9
 - lint, 139
 - names and locations, 165–167
 - networking, shared objects and, 165
 - profiled, 392
 - resource limits, 164, 166
 - search path rules, 152
 - search path specification, 152
 - shared
 - building, 151, 154
 - changes in, 164
 - version numbering, 153, 154, 164
 - table of routines, 287, 379
- Library Routine reference table, 287, 379
- libsocket directory, 178
- libw directory, 179
- /libxpg directory, 180
- /libxpg2.a library, 180
- limits, xiii
 - resource, 164, 166
- line command, 225
- line library routine, 323
- linemod library routine, 323
- link command, 225
- link system call, 268
- link system file, 383
- linkers, 148, 155
 - building executables, 151
 - building shared libraries, 151, 154
 - dynamic linking, 9
 - examples, 154, 155
 - library search path rules, 152
 - library search path specification, 152
 - link editor option changes, 148, 151
 - version numbering, 153, 154
- lint command, 139, 175, 225
- lint libraries, 139
- listen port monitor, 23, 114, 225
- listen system call , 268
- listing
 - file names and statistics, 79
 - file systems to save, 29
 - mounted resources, 397
 - shareable resources, 397
 - shared file systems, 397
 - software packages installed, 13
- ln command, 226
- loading device drivers, 9, 18, 184, 198
- loadkeys command, 226
- local commands directory, 393
- /local directory, 393
- /local.cshrc file, 47
- /local.login file, 48
- /local.profile file, 48
- localdtconv library routine, 323
- locale database, 383, 392
- localeconv library routine, 324
- localization databases, 392
- localtime library routine, 324, 362
- lockd command, 226
- lockf library routine, 324
- locks
 - condition variables, 195
 - master, 195
 - multithreaded-style, 195
 - mutexes, 157, 195
 - network lock daemon, 397
 - spooling lock files directory, 390
- /locks directory, 390
- lockscreen command, 226
- LOFS (loopback file system), 78

- log files
 - cron, 389
 - Service Access Facility (SAF), 390
 - system, 389
 - uucp, 121, 391
- log library routine, 324
- log10 library routine, 324
- log1p library routine, 324
- log2 library routine, 324
- logb library routine, 324
- logger command, 226
- logical device names, 188
- login command, 226
- login file, 40
- .login file, 47
- login shells
 - default home directory startup files, 48, 53
 - features, 46, 53
 - initialization files, 47
 - restricted, 41
 - selecting default, 45, 46
- logins, administering, 24, 40
- logname command, 226
- logname library routine, 181
- longjmp library routine, 324
- _longjmp library routine, 289
- longname library routine, 325
- look command, 226
- lookbib command, 226
- loopback file system (LOFS), 78
- lorder command, 172, 226
- lp command, 22, 110, 112, 226, 395
- /var/spool/lp directory, xiii
- /lp directory, , *see* /etc/lp directory
- lpadm command, 111, 227
- lpc command, 110, 111, 227
- lpd command, 110, 227, 397
- lpmove command, 112
- lpq command, 22, 110, 227, 395
- lpr command, 22, 110, 227, 395
- lprm command, 22, 110, 111, 227, 395
- lprof command, 246
- lpsched command, 397
- lpstat command, 22, 110, 227, 395
- lpsystem command, 111

- lptest command, 227
- lrand48 library routine, 325
- ls command, 93, 227
- lsearch library routine, 325
- lseek system call, 268
- lstat system call, 268
- lsw command, 227
- ltol3 library routine, 325
- lwp_checkstkset library routine, 325
- lwp_create library routine, 325
- lwp_ctxinit library routine, 325
- lwp_ctxmemget library routine, 325
- lwp_ctxmemset library routine, 325
- lwp_ctxremove library routine, 325
- lwp_ctxset library routine, 325
- lwp_datastk library routine, 325
- lwp_destroy library routine, 326
- lwp_enumerate library routine, 326
- lwp_errstr library routine, 326
- lwp_fpset library routine, 326
- lwp_geterr library routine, 326
- lwp_getregs library routine, 326
- lwp_getstate library routine, 326
- lwp_join library routine, 326
- lwp_libcset library routine, 326
- lwp_newstk library routine, 326
- lwp_perror library routine, 326
- lwp_ping library routine, 326
- lwp_resched library routine, 326
- lwp_resume library routine, 326
- lwp_self library routine, 326
- lwp_setpri library routine, 326
- lwp_setregs library routine, 326
- lwp_setstkcache library routine, 326
- lwp_sleep library routine, 326
- lwp_stkcswwset library routine, 327
- lwp_suspend library routine, 327
- lwp_yield library routine, 327

M

- m4 command, 172, 227
- m68k command, 227
- mach command, 227
- macros
 - kadb, 156, 157

- nroff, 394
- troff, 394
- madd library routine, 327
- madvise library routine, 327
- magic system file, 383
- magnetic tape devices, , *see* tape devices\x0d
- mail
 - auxiliary programs directory, 392
 - configuration information directory, 388
 - directory for, 390, 397
 - multimedia, 4
 - queued directory, 390
 - using, 50, 51
- mail administration, 20
- Mail command, 202
- mail command, xiii, 50, 51, 207, 227
 - /var/spool/mail directory, xiii
- Mail Tool (OpenWindows), 228
 - /mail directory, , *see* /etc/mail directory
- mailbox spooling directory, 20
- mailrc_to_defaults command, 228
- mailstat command, 228
- mailtool interface, 51, 228
- mailx command, 51, 202, 227, 393
 - /mailx directory, 393
- mailx program, 20
- make command, 167, 168, 172, 228
- makedbm command, 228
- makedev command, 228
- MAKEDEV environment variable, 68
- Makefiles, 167, 168
- makekey command, 228
- malloc library routine, 181, 327
- malloc_debug library routine, 327
- malloc_verify library routine, 327
- malloccmp library routine, 327
- man command, 53, 54, 229
 - /man directory, 51, 54
- man pages
 - directory organization changes, 51, 53
 - notation conventions, xvii, xviii
 - whatis database, 54, 204
 - windex database, 54, 204
- man.cf files, 53
- man68010 command, 229
- man68020 command, 229
- man68881version command, 229
- MANPATH environment variable, 53
- MANSECTS environment variable, 54
- maps (NIS), , *see* NIS (Network Information Services)\x0d
- maps (NIS+), , *see* NIS+ (Network Information Services Plus)\x0d
- master lock, 195
- master servers, , *see* NIS (Network Information Services), master servers\x0d
- matherr library routine, 327
- max_normal library routine, 327
- max_subnormal library routine, 327
- maximums, , *see* limits\x0d
- mblen library routine, 327
- mbstowcs library routine, 327
- mbtowc library routine, 327
- mcmp library routine, 327
- mconnect command, 229
- mcs command, 174
- mctl system call, 268
- mdiv library routine, 327
- memalign library routine, 327
- memccpy library routine, 327
- memchr library routine, 328
- memcmp library routine, 294, 328
- memcntl system call, 268
- memcpy library routine, 294, 328
- memset library routine, 295, 328
- menus, sysadm menu utility, 10
- merging files, 32, 33
- mesg command, 229
- message catalogs, 179
- messages, spoken, 5
- meta library routine, 328
- metadb command, 28
- metadevice configuration information, saving, 28
- metastat command, 28
- mfree library routine, 328

- min library routine, 328
- min_normal library routine, 328
- min_subnormal library routine, 328
- mincore system call, 268
- /misc directory, 89
- mkdir command, 229
- mkdir system call, 269
- mkfifo system call, 269
- mkfile command, 229
- mkfs command, 80, 94, 229
- mknod command, 229
- mknod system call, 270
- mkproto command, 229
- mkstemp library routine, 328
- mkstr command, 229
- mktemp library routine, 328
- mktime library routine, 362
- mlock library routine, 329
- mlockall library routine, 329
- mmap system call, 270
- /mnt directory, 389
- mnttab file, 397
- moddebug macro, 157
- moddir variable, 18, 184
- modems, managing, 23, 113, 115
- modf library routine, 329
- modifying, , *see* changing\x0d
- modinfo command, 183, 230
- modload command, 18, 184, 186, 198, 229
- modstat command, 230
- modules, , *see* kernel
- modunload command, 18, 184, 198, 230
- mon_break library routine, 329
- mon_cond_enter library routine, 329
- mon_create library routine, 329
- mon_destroy library routine, 329
- mon_enter library routine, 329
- mon_enumerate library routine, 329
- mon_exit library routine, 329
- mon_waiters library routine, 329
- monacct command, 230
- moncontrol library routine, 329
- MONITOR library routine, 290
- monitor library routine, 330
- monitoring file systems, 92
- monstartup library routine, 330
- more command, 230
- mount command, 80, 91, 92, 230, 396
- mount system call , 271
- mount_tfs command, 230
- mountall command, 80, 396
- mountd command, 230
- mounting
 - file systems, 18, 34, 80, 90, 92, 230, 271, 396
 - automounting, 18, 90–92, 206
 - mount table, 397
 - listing mounted resources, 397
 - remote resources, 80, 90
- mout library routine, 330
- move library routine, 331
- mprotect system call, 272
- /mqueue directory, 390
- mrnd48 library routine, 331
- msg_enumrecv library routine, 331
- msg_enumsend library routine, 331
- msg_recv library routine, 331
- MSG_RECVALL library routine, 290
- msg_reply library routine, 331
- msg_send library routine, 331
- msgctl system call, 272
- msgfmt command, 180, 223
- msgget system call , 272
- msgrcv system call , 272
- msgsnd system call , 272
- msub library routine, 331
- msync library routine, 331
- msync system call , 272
- MT (multithreaded) kernel, 9, 195
- mt command, 230
- MT-style locks, 195
- mtab system file, 383, 397
- mtio ioctls, 159, 161
- mtox library routine, 331
- mult library routine, 331
- multimedia mail, 4
- multiple OS operation, 102
- multithreaded (MT) kernel, 9, 195
- multithreaded (MT), defined, 195
- multiuser run levels, 69
- multiuser systems, shutting down, 70, 71

- munlock library routine, 331
- munlockall library routine, 331
- munmap system call , 272
- mutex macro, 157
- mutexes, 157, 195
- mv command, 230
- mvaddch library routine, 332
- mvaddstr library routine, 332
- mvcur library routine, 332
- mvdelch library routine, 332
- mvgetch library routine, 332
- mvgetstr library routine, 332
- mvinch library routine, 332
- mvinsch library routine, 332
- mvprintw library routine, 332
- mvscanw library routine, 332
- mvwaddch library routine, 333
- mvwaddstr library routine, 333
- mvwdelch library routine, 333
- mvwgetch library routine, 333
- mvwgetstr library routine, 333
- mvwin library routine, 333
- mvwinch library routine, 333
- mvwinsch library routine, 333
- mvwprintw library routine, 333
- mvwscanw library routine, 334

N

- name service request, foreign, entry points for, 393
- Name Service Switch, 22, 124, 178
- name services, , *see* DNS (domain name system)
- named command, 230
- NAMEFS (name file system), 74, 78
- names
 - device
 - CD-ROMs, 59
 - changes in, 15, 57, 59
 - developer's perspective, 187, 189
 - disks, 58, 59, 188
 - instance, 189
 - logical, 188
 - physical, 188
 - directories, 73
 - disks attached to system, 27

- explicit, for opening files using file descriptors, 74
- file, 32, 73, 79
- kernel name, 16
- libraries, 165–167
- napms library routine, 334
- nawk command, 230
- ncheck command, 80, 212, 230
- ndbootd command, 230
- neqn command, 230
- /usr/net/servers directory, xiii
- /net directory, , *see* /etc/net directory
- net_addr library routine, 334
- netdir_getbyname library routine, 314
- netgroup system file, 34, 384
- netmasks database, 384
- netname2host library routine, 334
- netname2user library routine, 334
- netrc system file, 384
- netstat command, 230
- /netsvc directory, 239, 392
- network devices, administering, 23
- Network Information Services Plus, , *see* NIS+ (Network Information Services Plus)\x0d
- Network Information Services, , *see* NIS (Network Information Services)\x0d
- Network Interface Tap (NIT), 178
- network lock daemon, 397
- networking libraries, shared objects and, 165
- networks
 - backups across, 97
 - booting over, 67, 185, 186
 - configuration information directory for transport-independent, 388
 - order of installing Solaris 2.6 on, 30
 - service administration, 23, 117, 122
 - Solaris 2.6 features, 177, 178
- networks database, 384
- newaliases command, 230
- newfs command, 95, 231

- newgrp command, 231
- newkey command, 231
- newpad library routine, 334
- /news directory, 390
- newterm library routine, 334
- newwin library routine, 334
- nextafter library routine, 334
- nextkey library routine, 334
- nexus drivers, 197
- nexus nodes, 197
- NFS
 - automounting of file systems shared through, 18
 - auxiliary programs and daemons directory, 392
 - binder process, 397
 - command changes, 118
 - daemons, 392, 398
 - listing mounted, 396
 - listing shared (exported), 396
 - mount request server, 397
 - secure, 39
 - support for, 77
- /nfs directory, 392, 397
- nfsd command, 231, 398
- nfsstat command, 231
- nfssvc system call, 272
- nfssys system call, 272
- nice command, 231
- nice library routine, 334
- nint library routine, 334
- NIS (Network Information Services), xiii
 - clients, restoring system data for, 32
 - master servers
 - restoring system data for, 32
 - saving system data for, 29
 - NIS+ migration, 126, 127
 - NIS+ vs., 124, 125
- /nis directory, 390
- NIS+ (Network Information Services Plus), xiii
 - database directory, 390
 - described, 4, 8, 21, 123, 124, 177
 - DNS vs., 124, 125
 - NIS migration to, 126, 127
 - NIS vs., 124, 125
 - planning an upgrade, 127
 - searching, 22
 - TCP/IP and, 117
 - updating, 22
- nispaswd command, 41
- NIT (Network Interface Tap), 178
- nl command, 231
- nl library routine, 334
- nl_init library routine, 334
- nl_langinfo library routine, 334
- nlist library routine, 335
- nlm_prot library routine, 335
- nlsadmin command, 23, 231
- nm command, 172, 232
- nocbreak library routine, 335
- nocrmode library routine, 335
- nodelay library routine, 335
- nodes, device tree, 197, 198
- noecho library routine, 335
- nohup command, 232
- nonl library routine, 335
- nonstandard_arithmetic library routine, 335
- noraw library routine, 335
- notation conventions, xvii, xviii
- notimeout library routine, 335
- nrand84 library routine, 335
- nroff command, 232, 392–394
- nslookup command, 232
- nsquery command, 232
- nsswitch.conf file, 178
- /nterm directory, 393
- ntohl library routine, 335
- ntohs library routine, 335
- nulladm command, 232

O

- /oasys directory, 393
- objdump command, 175
- object code disassembler, for COFF, 174
- od command, 232
- /old directory, 87, 173, 393
- old-analyze command, 232
- old-cat command, 232
- old-clocktool command, 232

- old-compact command, 232
- old-eyacc command, 232
- old-filemerge command, 232
- old-make command, 232
- old-perfmon command, 232
- old-prmail command, 232
- old-pti command, 233
- old-setkeys command, 233
- old-sun3cvt command, 233
- old-syslog command, 233
- old-uncompact command, 233
- old-vc command, 233
- OLIT (OPEN LOOK Intrinsic Toolkit), 171
- on command, 233
- on_exit library routine, 335
- OPEN LOOK admintool, 4
- OPEN LOOK Intrinsic Toolkit (OLIT), 171
- open system call , 273
- opendir library routine, 336
- opening files, using file descriptors, 74
- openlog library routine, 336
- openpl library routine, 336
- /openwin directory, 393
- OpenWindows, xiii
 - clock command, 209
 - Command Tool, 209
 - Developer's Guide File Chooser vs. XView File Chooser, 49
 - File Manager changes, 24, 62
 - Icon Edit tool, 221
 - indent command, 203
 - mail interface, 50
 - Mail Tool, 228
 - mount or installation point, 393
 - multiple displays, 203
 - Performance Meter tool, 234
 - PrintTool, 22
 - property sheets, 222
 - property window, 210, 212, 228, 242
 - Shell Tool, 242
 - SunView replaced by, 245
 - Text Edit tool, 247
 - version differences, 48
 - xlock command, 226

- xset command, 241
- OpenWindows, *see* Administration Tool\x0d
 - Administration Tool, "Operation not applicable for FSType " message, 80
 - /opt directory, 84, 89, 389
 - /opt file system, 75, 76
 - /opt, 75
 - /opt directory, xiii
 - /opt/sunwspro directory, 389
- optarg library routine, 336
- optind library routine, 336
- optional packages, directories for, 388, 390
- /options directory, 390
- OSF/Motif, 130
 - CDE compliance with, 130
- overlay library routine, 337
- overview command, 233
- overwrite library routine, 337

P

- pac command, 233
- pack command, 233, 250
- packages
 - adding, 170
 - administering, 393
 - advantages, 12, 26
 - compatibility packages, 35
 - components, 169
 - creating, 170
 - defined, 12, 169
 - developers and, 169, 170
 - information reporting, 13, 170
 - installing, 12, 13
 - optional, 388, 390
 - removing, 12, 13, 170
 - spooled, directory for, 390
- page command, 233
- pagesize command, 233
- passwd command, 41, 233
- passwd file, 34, 40, 384, 397
- passwd.adjunct system file, 384
- passwd2des library routine, 337
- passwords

- aging, 40, 41
- changing, 40, 41
- location of, 34, 40, 384, 397
- paste command, 234
- path names, xiii
 - generating for file systems, 80
- pathconf system call, 273
- pause library routine, 337
- pax command, 234
- paxcpio command, 234
- pcat command, 234
- PCFS (PC file system), 77
- pclose library routine, 337
- pdpll command, 234
- pechochar library routine, 337
- percent sign (%)
- perfmeter command, 234
- performance log file (uucp), 121
- Performance Meter tool
 - (OpenWindows), 234
- perror library routine, 338
- personal productivity tools, 4, 7
- pg command, 234
- phones database, 384
- physical device names, 188
- PIC archives directory, 392
- /pics directory, 392
- ping command, 234
- pipe files, pseudo file system for, 74
- pipe system call, 273
- /pkg directory, 390
- pkgadd command, 12, 13, 170, 217
- pkgask command, 170
- pkgchk command, 13, 170
- pkginfo command, 13, 170
- pkginfo file, 169
- pkgmk command, 170
- pkgparam command, 171
- pkgproto command, 170
- pkgrm command, 12, 13, 170
- pkgtrans command, 170
- plock library routine, 338
- plot command, 234
- plot library routine, 338
- plot system file, 384
- plottoa command, 234

- Pluggable Authentication Module (PAM), 43
- pmadm command, 23, 24, 113, 114
- pmap_getmaps library routine, 338
- pmap_getport library routine, 315, 338
- pmap_rmtcall library routine, 338
- pmap_set library routine, 338
- pmap_unset library routine, 338
- pnoutrefresh library routine, 338
- pnp library routine, 338
- pod_getexit library routine, 338
- pod_getmaxpri library routine, 338
- pod_getmaxsize library routine, 338
- pod_setexit library routine, 338
- pod_setmaxpri library routine, 338
- point library routine, 338
- point-to-point protocol (PPP), 118
- poll system call, 273
- popen library routine, 339
- port monitor services, administering, 23, 113, 114
- port monitors
 - administering, 23, 113, 114
 - described, 114
- portability, 5
- porting applications, reasons for, 35
- porting device drivers, 195, 196
- portmap command, 234
- PostScript filters, 51
- pound sign (#)
- pow library routine, 339
- PPP (point-to-point protocol), 118
- pr command, 234
- praudit command, 234
- prctmp command, 234
- prdaily command, 234
- prefresh library routine, 339
- /preserve directory, 390
- print subsystem
 - backend executables directory, 392
 - configuration information
 - directory, 388
 - database directory, 392
 - described, 22, 23
 - logging information directory, 390

printcap database, , *see* /etc/printcap
 database\x0d
 printenv command, 234
 printers, 109, 112
 canceling print jobs, 22, 111, 207, 395
 changes in, 22, 110
 commands, xiii
 changes in, 22
 using, 110, 112
 configuration, 34, 84
 daemon file, 397
 network requests for services,
 administering and
 troubleshooting, 23
 setting up
 commands, 110, 112, 395
 spool files directory, 390
 status checking, 110, 395
 printf library routine, 339
 PrintTool, 22
 printw library routine, 340
 priocntl command, 237, 392
 priocntl system call, 278
 priocntl system call, 266
 prioritizing, Solaris 2.6 installation for
 networks, 30
 /proc directory, 74
 /proc file system, 76
 process management, xiii
 pseudo file systems for, 74
 utilities, 8
 PROCFS (process access file system), 74,
 78
 productivity tools, , *see* DeskSet\x0d
 prof command, 172, 235
 prof library routine, 340
 profil library routine, 329, 330
 profil system call , 274
 .profile file, 47
 profiled libraries, 392
 programming tools, , *see* developers
 PROM, booting from, 67
 prompts
 property sheets (OpenWindows), 222
 property window (OpenWindows), 210,
 212, 228, 242

proto system file, 384
 protocols database, 384
 prototype file, 169
 prs command, 172, 235
 prt command, 172, 235
 prtacct command, 235
 prtconf command, 61, 194, 218
 prtvtoc command, 59, 60, 214
 ps command, 235, 396
 pscat (C/A/T) filters, 23, 51
 pseudo file systems, 74, 75, 78
 pseudo terminal (pty) slave devices,
 directory for, 83, 387
 pseudo-device driver modules, directory
 for, 89
 psignal library routine, 340
 pstat command, 235, 396
 ptrace request values, 162, 163
 ptrace system call , 274
 /pts directory, 83, 387
 ptx command, 235
 /pub director, 393
 publickey database, 384
 putc library routine, 340
 putchar library routine, 340
 putenv library routine, 340
 putmntent library routine, 291
 putmsg system call , 274
 putp library routine, 340
 putpwent library routine, 340
 puts library routine, 341
 putw library routine, 341
 pwck command, 235
 pwd command, 236
 pwauth library routine, 341
 pwauthd command, 236

Q

qsort library routine, 341
 queuedefs system file, 384
 quick reference table, 395, 398
 QuickCheck, 27
 quiet_nan library routine, 341
 quot command, 93, 236
 quota command, 236

quotacheck command, 236
quotactl system call, 274
quotaoff command, 236
quotaon command, 236

R

rand library routine, 341
random library routine, 341
ranlib command, 173, 236
rarpd command, 236
rasfilter8tol command, 236
raster image filters, 23, 51
rasterfile system file, 384
rastrel command, 236
raw disk devices, directory for, 82, 188, 387
raw library routine, 341
raw tape devices, directory for, 83, 387
rc scripts, 68–70, 74, 83, 87, 186, 236, 388, 398
rc.boot script, 68, 69, 83, 186, 236
rc.d scripts, 83, 388, 398
rc.local script, 68, 69, 83, 186, 236, 398
rc.single script, 68, 83, 186
rcmd library routine, 341
rcp command, 236
rcS script, 68, 70, 74, 84, 87, 186
rcS.d script, 84, 388, 398
rdate command, 236
rdist command, 237
/rdsk directory, 82, 83, 188, 387
rdump command, 237
re_comp library routine, 343
re_exec library routine, 343
read system call, 274
readdir library routine, 342
readlink system call, 275
readv system call, 275
real-time priority scheduling, 4
realloc library routine, 343
realpath library routine, 343
reboot command, 66, 70, 72, 237
reboot system call, 275
reconfiguration boot, 186
recv system call, 275
recvfrom system call, 276

recvmsg system call, 276
red command, 237
refer command, 237
/refer directory, 392
refresh library routine, 343
regcmp command, 174
regexp library routine, 343
registerrpc library routine, 343
regular expression compiler, 174
rehash command, 237
reject command, 112
rem_drv command, 186, 198
remainder library routine, 343
remexportent library routine, 343
remote CD-ROM devices, install4x program and, 104
remote file systems
 automounting, 18
 default type, 81
 mounting, 80, 90
 unmounting, 80
remote procedure calls, administering, 24
remote system file, 384
remote systems
 software package administration, 12, 13
remote tape drives, backing up to, 97
remove_brackets command, 237
removing
 devices, 186, 198
 network devices, 23
 NIS+ information, 22
 port monitor services, 114
 port monitors, 114
 software packages, 12, 13, 170
remque library routine, 344
rename system call, 276
renice command, 237
repquota command, 237
res_init library routine, 344
res_mkquery library routine, 344
res_send library routine, 344
reset command, 237
reset_prog_mode library routine, 309, 344
reset_shell_mode library routine, 344

resetterm library routine, 344
 resetty library routine, 344
 resolv.conf system file, 384
 resource limits, 164, 166
 restartterm library routine, 344
 restore command, xiii, 96, 237, 396
 restoring, xiii, 34
 file systems, 97
 SunOS release 4.x system data, 34
 restoring files and file systems, 31
 ufsrestore command, xiii
 rev command, 237
 rewind library routine, 344
 rewinddir library routine, 344
 rex library routine, 344
 rexd command, 237
 rexec library routine, 344
 rexecd command, 237
 rfadmin command, 237
 rfmaster system file, 384
 rfpaswd command, 237
 /RFS file system, 74
 rfstart command, 237
 rfstop command, 237
 rfudadmin command, 238
 rfudaemon command, 238
 rgb system file, 384
 .rhosts files, 39
 rhosts system file, 384
 .rhosts.equiv files, 39
 rindex library routine, 344
 ring_alarm command, 238
 rint library routine, 344
 ripoffline library routine, 344
 rksh command, 41
 rlogin command, 238
 rlogind command, 238
 rm command, 238
 rm_client command, 238
 rm_services command, 238
 rmail command, 238
 rmdel command, 173, 238
 rmdir system call, 276
 rmdir command, 239
 rmmount command, 63
 rmmount.conf file, 63
 rmnstat command, 239
 rmt command, 239
 /rmt directory, 83, 387
 rmtab system file, 384
 rnusers library routine, 344
 roffbib command, 239
 root access, defaults, 40
 root file system (/), 29, 76, 387, 391
 rootmenu system file, 385
 route command, 239
 routed command, 239
 rpc database, 385
 RPC, secure, 39, 43
 rpc.bootparamd command, 239
 rpc.etherd command, 239
 rpc.lockd command, 239, 397
 rpc.mountd command, 239, 397
 rpc.rexd command, 239
 rpc.rquotad command, 239
 rpc.rstatd command, 239
 rpc.rusersd command, 239
 rpc.rwalld command, 239
 rpc.showfhd command, 239
 rpc.sprayd command, 239
 rpc.statd command, 240
 rpc.user_agentd command, 240
 rpc.yppasswdd command, 240
 rpc.yppupdated command, 240
 rpc_broadcast library routine, 296
 rpc_call library routine, 295
 rpc_createerr library routine, 344
 rpcb_getaddr library routine, 338
 rpcb_getmaps library routine, 338
 rpcb_rmtcall library routine, 338
 rpcb_set library routine, 338
 rpcb_unset library routine, 338
 rpcbind command, 234
 rpcgen command, 240
 rpcinfo command, 240
 rpow library routine, 345
 rquota library routine, 345
 rrestore command, 240
 rresvport library routine, 345
 rsh command, 41, 240
 rstat library routine, 345

- rtime library routine, 345
- run levels, , *see* rc scripts\x0d
- run-state transition operations, directory
 - for, 84
- runacct command, 240
- rup command, 240
- ruptime command, 240
- rusage command, 240
- ruserok library routine, 345
- rusers command, 240
- rusers library routine, 345
- rwall command, 240
- rwall library routine, 345
- rwho command, 240

S

- s.files, 168
- S5 (System V file system), 79
- sa command, 240
 - /sa directory, 392
- SAC (Service Access Controller), 23, 24
- sacadm command, 23, 114
- sact command, 173, 240
 - /sad directory, 83, 387
 - /var/sadm directory, xiii
 - /sadm directory, , *see* /usr/sadm directory
 - /var/saf directory, xiii
 - /saf directory, , *see* /etc/saf directory
- SAMECV library routine, 290
- SAMEMON library routine, 290
- SAMETHREAD library routine, 290
- sar command, 202, 235
- savecore command, 241
- saveterm library routine, 345
- savetty library routine, 345
- saving, xiii
 - disk partition information, 27
 - file system information, 27
 - metadevice configuration information, 28
- /sbin directory, xiii
 - changes in, 74
 - described, 76, 82, 87, 389
- /sbin/init command, 66–68, 70
- /sbin/rc scripts, 69, 70, 74, 83, 87
 - /sbin/rc scripts, xiii
 - /sbin/rc scripts, 69, 70, 74, 83, 87
 - /sbin/rcS scripts, 70, 84, 87
 - /sbin/rcS scripts scripts, 74
- sbrk library routine, 181
- sbrk system call, 276
- scalability, 5, 9
- scalb library routine, 345
- scalbn library routine, 345
- scandir library routine, 346
- scanf library routine, 346
- scanw library routine, 346
- SCCS (source code control system), 168
- sccs command, 173, 241
 - /sccs directory, 168, 172
 - sccs-admin command, 241
 - sccs-cdc command, 241
 - sccs-comb command, 241
 - sccs-delta command, 241
 - sccs-get command, 241
 - sccs-help command, 241
 - sccs-prs command, 241
 - sccs-prt command, 241
 - sccs-rmdel command, 241
 - sccs-sact command, 241
 - sccs-sccsdiff command, 241
 - sccs-unget command, 241
 - sccs-val command, 241
 - /sccs directory, 173
- sccsdiff command, 173, 241
- sccsfile system file, 385
- /sched directory, 89
- scheduling, , *see* Calendar Manager
- scr_dump library routine, 346
- scr_init library routine, 346
- scr_restore library routine, 346
- screenblank command, 241
- screendump command, 241
- screenload command, 241
- script command, 242
- scripts
 - accounting, 392
 - FMLI, binaries directory for, 393
 - package management, 393

- profile scripts for new user
 - accounts, 389
- rc, 68–70, 74, 83, 87, 186, 236, 388, 398
- software package, 170
- spell command, 393
- system activity report package, 392
- scroll library routine, 346
- scrolldefaults command, 242
- scrollok library routine, 347
- SCSA (Sun common SCSI architecture),
 - DDI only interfaces, 193
- SCSI disks
 - multithreaded kernel and, 196
 - naming conventions, 58
 - Sun common SCSI architecture (SCSA), 193
- sd1g command, 68
- sdiff command, 242
- search paths
 - kernel modules, 18, 184
 - linker, 152, 153
- searching
 - for files, 93
 - NIS+ tables, 22
- seconvert library routine, 347
- security, xiii, 39
 - ASET, 8, 42
 - changes in, 39, 204
 - features, 39
 - Kerberos, 43
 - level of, 8, 42
 - overview, 39
 - PAM, 43
 - restricted shells, 41
 - SunShield, 43
 - unbundled, 43
- security log file (uucp), 121
- sed command, 242
- seed48 library routine, 347
- seekdir library routine, 347
- select system call, 276, 365
- selection_svc command, 242
- semctl system call, 276
- semget system call, 276
- semop system call, 276
- send system call, 276
- sendmail command, 20, 238, 242
- sendmail.cf file, 34
- sendmsg system call, 277
- sendto system call, 277
- Serial Port Manager (Administration Tool), 113
- Serial Port Manager (admintool), 113
- serial ports, controlling, 23, 34
- servers, xiii
 - /servers directory, 393
- Solaris 2.6 server, xiii
- Service Access Controller (SAC), 23, 24
- Service Access Facility (SAF), 23, 113, 115
 - auxiliary programs and daemons
 - directory, 392
 - commands, 23, 114
 - described, 23, 113, 115
 - directory for, 84, 389
 - logging and accounting file
 - directory, 88, 390
 - Service Access Controller (SAC)
 - and, 23, 24, 113
- services system file, 385
- set4 command, 242
- set_alarm command, 242
- set_curterm library routine, 347
- set_term library routine, 349
- setac library routine, 347
- setaudit system call, 277
- setaudit system call, 277
- setbuf library routine, 347
- setbuffer library routine, 347
- setdomainname system call, 277
- setegid library routine, 347
- setegid system call, 278
- seteuid library routine, 347
- seteuid system call, 278
- setexportent library routine, 347
- setfsent library routine, 347
- setgid library routine, 347, 348
- setgid system call, 278
- setgraent library routine, 347
- setgrent library routine, 347
- setgroups system call, 277
- sethostent library routine, 347

- sethostname system call, 277
- setitimer system call, 277
- setitimer system call, 364, 365
- setjmp library routine, 347
- _setjmp library routine, 289
- setkey library routine, 348
- setkeys command, 242
- setlinebuf library routine, 348
- setlocale command, 181
- setlocale library routine, 348
- setlogmask library routine, 348
- setmntent library routine, 348
- setnetent library routine, 348
- setnetgrent library routine, 348
- setpgid system call, 277
- setpgrp system call, 278
- setpriority system call, 278
- setprotoent library routine, 348
- setpwaent library routine, 348
- setpwent library routine, 348
- setpwfile library routine, 348
- setregid system call, 278
- setreuid system call, 278
- setrgid library routine, 348
- setrlimit system call, 278
- setrpercent library routine, 348
- setruid library routine, 348
- setscreg library routine, 348
- setservent library routine, 348
- setsid command, 242
- setsid system call, 278
- setsockopt system call, 279
- setstate library routine, 348
- setsyx library routine, 348
- setterm library routine, 349
- settimeofday system call, 279
- settyent library routine, 349
- setuid library routine, 349
- setuid system call, 278
- setuid system call, 348
- setup_client command, 242
- setup_exec command, 242
- setupterm library routine, 349
- setuseraudit system call, 279
- setusershell library routine, 349
- setvbuf library routine, 349
- sfconvert library routine, 349
- sgconvert library routine, 349
- sgetl system call, 279
- sh command, 45–47, 242
- shadow file, 34, 40, 86, 397
- share command, 91, 94, 217, 396
- /share directory, , see /usr/share directory\x0d
- /share file system, mounting, 90
- shareable resources, listing, 397
- shareall command, 94, 396
- shared file systems
 - configuration information
 - directory, 388
 - default root of, 389
 - described, 93
 - listing, 397
 - mounting, 90
 - Solaris 2.6 server for SunOS release
 - 4.x diskless/dataless
 - clients, 101
 - transitioning, 29, 34
- shared libraries
 - building, 151, 154
 - changes in, 164
 - version numbering, 153, 154, 164
- shared objects, networking libraries
 - and, 165
- sharetab file, 291, 397
- Shell Tool (OpenWindows), 242
- shells, xiii
 - default home directory startup
 - files, 48, 53
 - features, 46, 53
 - initialization files, 47
 - restricted, 41
 - selecting default, 45, 46
- C shell, xiii
- shelltool command, 242
- shift_lines command, 243
- shmat system call, 279
- shmctl system call, 279
- shmdt system call, 279
- shmget system call, 279
- showfh command, 243

showfhd command, 239
 showmount command, 92, 243, 396
 shutacct command, 243
 shutdown command, 66, 70, 71, 243
 shutdown system call, 279
 shutting down

- fastboot command, 66, 72, 217
- fasthalt command, 68, 70, 72, 217
- halt command, 66, 70, 72, 220
- reboot command, 66, 70, 72, 237
- shutdown command, 66, 70, 71
- shutdown system call, 279

 sigaction system call, 279
 sigaction library routine, 349
 sigaction system call, 280
 sigaddset library routine, 349
 sigaltstack system call, 280
 sigblock system call, 279
 sigdelset library routine, 349
 sigemptyset library routine, 349
 sigfillset library routine, 350
 sigfpe library routine, 350
 siginterrupt library routine, 350
 sigismember library routine, 350
 siglongjmp library routine, 289, 350
 sigmask system call, 279
 signal library routine, 350
 signaling_nan library routine, 350
 signbit library routine, 350
 significand library routine, 350
 sigpause system call, 279
 sigpending system call, 280
 sigprocmask routine, 280
 sigprocmask system call, 280
 sigsetjmp library routine, 289, 350
 sigsetmask system call, 280
 sigsetops routines, 279
 sigstack system call, 280
 sigsuspend system call, 280
 sigvec system call, 280
 sin library routine, 351
 single-user run levels, 69, 70
 single-user systems, shutting down, 68
 single_precision library routine, 351
 single_to_decimal library routine, 351
 sinh library routine, 351
 size command, 173, 243
 /skel directory, 47, 389
 skyversion command, 243
 sleep command, 195, 243
 sleep library routine, 351
 slices, , *see* disk slices\x0d
 slk_clear library routine, 351
 slk_init library routine, 351
 slk_label library routine, 351
 slk_noutrefresh library routine, 352
 slk_refresh library routine, 352
 slk_restore library routine, 352
 slk_set library routine, 352
 slk_touch library routine, 352
 slow devices, improving performance, 74, 78
 /sm directory, 389
 sm system file, 385
 /sm.bak directory, 389
 sm_inter library routine, 352
 /snadm directory, 87, 394
 snoop command, 216
 socket system call , 280
 socketpair system call, 280
 sockets, 178, 280
 sockio ioctls, 159, 161
 soelim command, 243
 software, xiii

- third-party and unbundled, 75, 76

 software clusters, , *see* clusters\x0d
 software groups, 26
 Software Manager, , *see* swmtool
 command\x0d
 software packages, , *see* packages
 Solaris 2.4 operating environment, xiii
 Solaris 2.6 DDI/DKI, 9, 192, 193, 198
 Solaris 2.6 DDI/DKI compliant,

- defined, 193

 Solaris 2.6 operating environment

- additional documentation on, xviii
- installation features, 25
- overview of major changes, 12, 24

 Solaris operating environment, xiii

- Solaris 2.6 server, SunOS release 4.x
 - diskless/dataless client support on, 101
- Solaris 2.6 SPARC DDI/DKI, 9, 192, 193, 198
- Solaris 2.x driver architecture, 197, 198
- Solaris CDE
 - application manager, 133
 - file manager, 132
 - front panel, 130
 - moving from OpenWindows, 133
 - style manager, 131
- Solaris Common Desktop Environment,,
 - see* Solaris CDE
- Solaris operating environment, xiii
 - advantages, 3, 6
 - compatibility, 4, 6
 - developer features, 8, 9
 - features, 3, 6, 9
 - interoperability, 5
 - large organizations and, 6
 - portability, 5
 - scalability, 5, 9
 - SVR4 vs., 3, 4, 6, 10
 - system administrator features, 7, 8
 - user features, 7
- Solstice DiskSuite, installation and, 28
- sort command, 243
- sortbib command, 243
- source code control system (SCCS), 168
- source code directory, 394
- space library routine, 352
- sparc command, 244
- SPARC DDI/DKI, 9, 192, 193, 198
- SPARC platforms, Solaris features and, 9
- SPARCServer Manager, installation
 - and, 28
- SPARCserver Manager, installation
 - and, 28
- SPECFS (device special file system), 78
- speed of devices, improving, 74, 78
- spell command, 244, 392, 393
 - /spell directory, 393
- spellin command, 244
- spline command, 244
- split command, 244
- splN/splr pairs, 195
- spoken messages, 4
- spool directory, , *see* /var/spool
 - directory\x0d
- spray command, 244
- spray library routine, 352
- sprint library routine, 352
- sputl system call , 280
- sqrt library routine, 352
- strand library routine, 352
- strand48 library routine, 352
- srandom library routine, 353
 - /src directory, 394
- sscanf library routine, 353
- ssignal library routine, 350, 353
- standard processes, quick reference, 397, 398
- standard_arithmetic library routine, 353
- standardization, 3–5
- standend library routine, 353
- standout library routine, 353
- startup, 65, 69
- startup command, 244
- stat system call, 280
- statfs system call, 280
- statically linked programs directory, 393
- statmon system file, 385
- status checking, , *see* checking
- status files, uucp, 391
- status monitor information directory, 389
- statvfs system call, 280
- STKTOP library routine, 290
- stopping system, , *see* shutting down\x0d
- store library routine, 353
- strcasecmp library routine, 353
- strcat library routine, 353
- strchr library routine, 318, 353
- strcmp library routine, 353
- strcoll library routine, 353
- strcpy library routine, 353
- strcspn library routine, 353
- strdup library routine, 353
- streamio ioctls, 159, 161
- STREAMS, xiii, 196, 199
 - changes in, 196

- described, 9
- Ethernet drivers and, 178
- ioctl transparency, 196, 197
- location of modules, 89
- pseudo file system for, 74
- pushing modules, 197
- Solaris 2.x driver architecture, 197, 198
- STREAMS Administrative Driver, entry
 - point directory, 83, 387
- streamtab structure, 197
- strftime command, 180
- strftime library routine, 354
- string_to_decimal library routine, 355
- strings command, 244
- strings, extracting from source files, 174
- strioctl structure, 196
- strip command, 173, 244
- strlen library routine, 355
- /strmod directory, 89
- strncasecmp library routine, 355
- strncat library routine, 355
- strncmp library routine, 355
- strncpy library routine, 355
- strpbrk library routine, 355
- strptime library routine, 355
- strchr library routine, 344, 355
- strspn library routine, 355
- strtod library routine, 355
- strtok library routine, 355
- strtol library routine, 355
- strxfrm library routine, 355
- stty command, 237, 244
- stty library routine, 355
- stty_from_defaults command, 244
- Style Manager, 134
- su command, 244
- su file, 40
- subpad library routine, 356
- subwin library routine, 356
- sum command, 245
- Sun C compiler, 139, 148
- sun command, 245
- Sun common SCSI architecture (SCSA),
 - DDI only interfaces, 193
- Sun DDI (Sun Device Driver Interface), 9, 192, 197
- Sun WorkShop Debugger, 156
- sundiag command, 245
- suninstall command, 245
- SunOS release 4.x
 - backing up file systems, 27, 30
 - Backup Copilot, 27
 - command changes, 35
 - compatibility with SunOS release 5.6, 35, 48
 - device naming and, 15
 - diskless/dataless clients, Solaris 2.6
 - server support for, 101
 - overview of major changes in Solaris 2.6, 12, 24
 - QuickCheck, 27
 - restoring system data, 34
 - saving disk partition information, 27
 - saving file system information, 27
 - saving metadvice configuration information, 28
 - tools, finding, 171
 - work environment, using with Solaris 2.6, 35, 48
- SunOS release 5.6
 - admintool, 8, 20, 21
 - automounting, 18
 - command changes, 35
 - compatibility packages, 35, 48
 - cross-functional compatibility, 4
 - device naming, 15
 - file systems, 15, 17
 - kernel configuration, 17
 - kernel name in, 16
 - print subsystem, 22, 23, 110
 - security, 39
- SunOS/BSD Source Compatibility Package, 35, 36
- SunShield Basic Security Module (BSM), 43
- SunView, 49, 391
- sunview command, 245
- sunview system file, 385
- /sunview1 directory, 391
- SUNWhinst, 102
- sv_acquire command, 245

sv_release command, 245
 sv_xv_sel_svc command, 242
 svc_create library routine, 357
 svc_destroy library routine, 356
 svc_dg_create library routine, 358
 svc_fd_create library routine, 357
 svc_fds library routine, 356
 svc_fdset library routine, 356
 svc_freeargs library routine, 356
 svc_getargs library routine, 356
 svc_getcaller library routine, 356
 svc_getreq library routine, 356
 svc_getreqset library routine, 356
 svc_raw_create library routine, 357
 svc_reg library routine, 357
 svc_register library routine, 357
 svc_run library routine, 357
 svc_sendreply library routine, 357
 svc_tli_create library routine, 357
 svc_unreg library routine, 357
 svc_unregister library routine, 357
 svc_vc_create library routine, 357
 svcerr_auth library routine, 357
 svcerr_decode library routine, 357
 svcerr_noproc library routine, 357
 svcerr_noprogram library routine, 357
 svcerr_progvers library routine, 357
 svcerr_systemerr library routine, 357
 svcerr_weakauth library routine, 357
 svcfcd_create library routine, 357
 svcraw_create library routine, 357
 svctcp_create library routine, 357
 svcudp_bufcreate library routine, 358
 svcudp_create library routine, 358
 svdtab system file, 385
 SVR4
 interchanging data, 98
 Solaris vs., 3, 4, 6, 10
 unsupported file system types, 79
 svr4.make command, 167, 168, 228
 swab library routine, 181, 358
 swap command, 235, 245, 396
 swap device, default, 74, 78
 SWAPFS (swap file system), 74, 78
 swapon command, 245
 swapon system call, 280
 swin command, 245
 switcher command, 245
 swmtool command
 add_services command vs., 203
 described, 12, 13
 extract_unbundled command vs., 217
 rm_services command vs., 238
 symlink system call, 281
 symorder command, 173, 245
 sync command, 246
 sync system call, 281
 syntax, file system commands, 80, 81
 /sys directory, 74, 89
 /sys, 16, 74
 sys_siglist library routine, 358
 sys_unconfig command, 246
 sysadm menu utility, 10
 syscall system call, 281
 sysconf system call, 264, 266, 281
 sysdef command, 59, 213, 221
 sysinfo system call, 263, 264, 277
 syslog library routine, 358, 367
 syslog.conf system file, 385
 syslogd command, 246
 system accounting
 directories for, 388, 389
 utilities overview, 8
 system activity report package, 392
 system administration directories, 393
 system administrators, xiii
 security procedures, 39
 Solaris features for, 7, 8
 , xiii
 System Calls reference table, 255, 285
 system command, 68
 system commands, directory for
 standard, 391
 system configuration
 changes in, 183, 186
 default, 84
 restoring data, 34
 system file, , see /etc/system file\x0d
 System Files reference table, 381, 386
 system header files, directory for, 196
 system information utilities, 8

system initialization scripts, , *see* rc scripts\x0d
system library routine, 358
system run levels, , *see* rc scripts\x0d
system scheduling, , *see* process management\x0d
System V file system (S5), 9, 79
System V Interface Definition, 163
systems system file, 385

T

t300 command, 246
t300s command, 246
t4013 command, 246
t450 command, 246
t_accept library routine, 358
t_alloc library routine, 358
t_bind library routine, 358
t_close library routine, 358
t_connect library routine, 358
t_error library routine, 358
t_free library routine, 358
t_getinfo library routine, 358
t_getstate library routine, 358
t_listen library routine, 358
t_look library routine, 359
t_open library routine, 359
t_optmgmt library routine, 359
t_rcv library routine, 359
t_rcvconnect library routine, 359
t_rcvdis library routine, 359
t_rcvrel library routine, 359
t_rcvudata library routine, 359
t_rcvuderr library routine, 359
t_snd library routine, 359
t_snddis library routine, 359
t_sndrel library routine, 359
t_sndudata library routine, 359
t_sync library routine, 359
t_unbind library routine, 359
tab setting escape sequences, directory for, 393
tabs command, 246
/tabset directory, 393
tail command, 246
talk command, 246

tan library routine, 359
tanh library routine, 359
tape devices
 backing up to remote, 97
 directory for raw, 83
tar command, xiii
 bar command vs., 206
 changes affecting, 98, 246
 described, 98
 support for, 96
 ustar command vs., 250
tar system file, 385
tbl command, 246
tcdrain library routine, 359
tcflow library routine, 359
tcflush library routine, 360
tcgetattr library routine, 360
tcgetpgrp library routine, 360
tcopy command, 246
tcov command, 175, 246
TCP/IP, 117
tcsendbreak library routine, 360
tcsetattr library routine, 360
tcsetpgrp library routine, 360
tdelete library routine, 360
tee command, 246
tek command, 246
tektool command, 246
tell system call, 281
telldir library routine, 360
telnet command, 246
template files, shell initialization, 47
tempnam library routine, 360
temporary file systems, xiii
 directory for files not cleared during boot sequence, 390
 directory for spooled temporary files, 390
 transitioning, 29
/term directory, 83, 388
term system file, 385
termcap database, 74, 385
terminal description files, directory for, 394
terminal devices, directory for, 83, 388

terminal tables, for nroff command, 393
terminals, managing, 23, 113, 115
terminfo database, 74, 110, 111, 385, 394,
397
termio interface, 317, 355
termio ioctls, 159, 161
termios ioctls, 159, 161
test command, 246
TEX filters, 23, 51
Text Edit tool (OpenWindows), 247
textdomain library routine, 181, 360
textedit command, 247
textedit_filters command, 247
tfind library routine, 360
TFS (translucent file system), 74, 78
tfsd command, 247
tftp command, 247
/tftpboot directory, 29
tgetent library routine, 360
tgetflag library routine, 361
tgetnum library routine, 361
tgetstr library routine, 361
tgoto library routine, 362
third-party software, file system for, 75, 76
thread macro, 156
threadlist macro, 156
tic command, 247
tigetflag library routine, 362
tigetnum library routine, 362
tigetstr library routine, 362
time command, 247
time library routine, 310, 362
time management, , *see* Calendar Manager
time zone information, 394
timegm library routine, 362
timelocal library routine, 362
times library routine, 362
timezone library routine, 362
tip command, 247
/tm directory, 389
/tmac directory, 394
/tmp directory, 389
/tmp file system (TMPFS)
described, 76
support for, 78
transitioning, 29
tmpfile library routine, 362
TMPFS, , *see* /tmp file system
(TMPFS)\x0d
tmpnam library routine, 362
toascii library routine, 363
toc system file, 385
tolower library routine, 363
_tolower library routine, 289
toolkits, 171
toolplaces command, 247
tools (developer), xiii, 159, 174
Binary Compatibility Package, 35, 36,
168
finding, 171, 174
ioctl requests, 159, 161
libraries, 163, 167
make command, 167, 168
OLIT, 171
ptrace request values, 162, 163
SCCS, 168
software packages, 169, 171
XView, 171
ToolTalk, 7
touch command, 247
touchline library routine, 363
touchoverlap library routine, 363
touchwin library routine, 363
toupper library routine, 363
_toupper library routine, 289
tparm library routine, 363
tput command, 247
tputs library routine, 364
tr command, 247
trace command, xiii, 157, 248
traceoff library routine, 364
traceon library routine, 364
trademark files, directory for, 389
traffic command, 248
translate system file, 385
translucent file system (TFS), 74, 78
troff command, 248
changes in, 23, 51, 112
macro packages for, 394
preprocessor for, 392
troubleshooting

- printing service requests from
 - network, 23
 - Service Access Controller (SAC), 23
 - TTY devices, 23
- trpt command, 248
- true command, 248
- truncate system call, 281
- truss command, 157, 174, 248
- tsearch library routine, 364
- tset command, 248
- tsort command, 173, 248
- tty command, 248
- TTY devices, administering and
 - troubleshooting, 23
- ttyadm command, 23
- ttymon port monitor, 23, 114
- ttyname library routine, 364
- ttyslot library routine, 364
- ttysoftcar command, 248
- ttytab system file, 34, 385
- tunefs command, 248
- turnacct command, 248
- tvconfig command, 248
- twalk library routine, 364
- typeahead library routine, 364
- types system file, 385
- typographic conventions, in manual, xvii, xviii
- tzfile system file, 385
- tzset library routine, 364
- tzsetup command, 248
- tzsetwall library routine, 364

U

- u370 command, 248
- u3b command, 248
- u3b15 command, 248
- u3b2 command, 248
- u3b5 command, 248
- uadmin system call, 275, 280
- ualarm library routine, 364
- /ucb directory, , *see* /usr/ucb directory
- /ucbinclude directory, 394
- /ucblib directory, 165, 167
- /ucblib directory, 394
- UFS, , *see* Unix file systems (UFS)\x0d
- ufsboot command, 67, 185, 186
- ufsbootblk command, 67
- ufsdump command
 - commands replaced by, 96, 215, 237, 396
 - described, 96, 97
- ufsrestore command
 - commands replaced by, 96, 237, 240
 - described, 97
 - quick reference, 396
- ul command, 248
- ulimit library routine, 365
- umask command, 249
- umask system call, 281
- umount command, 80, 91, 249, 396
- umount system call, 281
- umount_tfs command, 249
- umountall command, 80, 91, 396
- unadv command, 249
- uname command, 204, 221, 227, 249
- uname system call, 281
- unbundled software, 75
- unbundled software, file system for, 75, 76
- uncompress command, 249
- unconfigure command, 249
- unctrl library routine, 365
- unexpand command, 249
- unget command, 173, 249
- ungetc library routine, 365
- ungetch library routine, 365
- unifdef command, 173, 249
- uniq command, 249
- units command, 249
- unix, 16, 75
- /unix directory, 26, 67, 89, 184
- Unix file systems (UFS), xiii
 - creating, 95
 - cylinder groups and, 14
 - support for, 77
- unix kernel name, xiii, 16, 75, 186
- Unix System V Release 4, , *see* SVR4\x0d
- Unix-to-Unix Copy (UUCP), xiii, 34, 119, 122
- unix2dos command, 249
- unlink system call, 281

- unlink command, 249
- unloading, device drivers, 18, 184, 198
- unmount system call, 281
- unmounting
 - file systems, 80, 91
 - remote resources, 80
- unpack command, 249
- unshare command, 92, 396
- unwhiteout command, 250
- update command, 250
- updaters system file, 386
- uptime command, 250
- user access, controlling, 50
- User Account Manager (Administration Tool)
 - password changes and, 41
- user and group administration, 49
- user directories, default root of, 389
- user environment administration, xiii, 45, 55
 - default shell selection, 45, 46
 - document tool usage, 51
 - mail usage, 50, 51
 - man command search path
 - customization, 53, 54
 - man page directory organization
 - changes, 51, 53
 - SunOS 4.x work environment used
 - with Solaris 2.6, 35, 48
 - user and group administration, 49
 - window system, 48
- user management utilities, 8
- user2netname library routine, 365
- useradd command, 49, 50
- userdel command, 49, 50
- usermod command, 49, 50
- users command, 250
- users, Solaris features for, 7
- usleep library routine, 365
- /usr directory
 - changes in, 87
 - described, 74, 87, 389
- /usr file system
 - described, 76
 - layout changes, 391, 394
 - mount point for, 389
 - transitioning, 29
- /usr/4lib directory, 391
- /usr/5bin directory, 87
- /usr/5include directory, 87
- /usr/5lib directory, 87, 165, 166
- /usr/bin directory
 - described, 79, 391
 - generic file system administrative
 - commands in, 79, 80
 - programming tools in, 172, 173
 - quick reference, 397
 - SunOS release 4.x files moved to, 87
- /usr/bin/csh command, 45–47, 211
- /usr/bin/ksh command, 45–47
- /usr/bin/mail command, 50, 51, 207, 227
- /usr/bin/mailx command, 51, 202, 227, 393
- /usr/bin/rksh command, 41
- /usr/bin/rsh command, 41, 240
- /usr/bin/sh command, 45–47, 242
- /usr/bin/sunview1 directory, 391
- /usr/ccs directory, 87, 391
- /usr/ccs/bin directory, 168, 171, 391
- /usr/ccs/bin/make command, 167, 168, 172, 228
- /usr/ccs/lib directory, 165, 167, 171, 172, 391
- /usr/ccs/lib directory, xiii
- /usr/ccs/lib/svr4.make command, 167, 168
- /usr/ccs/lib/svr4.make command, 228
- /usr/ccs/libp directory, 166
- /usr/demo directory, 391
- /usr/etc directory, 87
- /usr/etc directory, xiii
- /usr/etc/biod command, 207, 398
- /usr/etc/nfsd command, 231, 398
- /usr/etc/rpc.lockd command, 239, 397
- /usr/etc/rpc.mountd command, 239
- /usr/etc/rpc.mountd command, 397
- /usr/etc/yplib command, 253, 397
- /usr/games directory, 391
- /usr/include directory, 87, 391
- /usr/kernel directory, , *see* /kernel directory\x0d

- /usr/kvm directory, 392
- /usr/lib directory, 87, 165, 166, 172, 173, 392
- /usr/lib directory, xiii
- /usr/lib/acct directory, 392
- /usr/lib/class directory, 392
- /usr/lib/dict directory, 392
- /usr/lib/font directory, 392
- /usr/lib/fs directory, 82, 83, 392
- /usr/lib/iconv directory, 392
- /usr/lib/libc directory, 180
- /usr/lib/libintl directory, 392
- /usr/lib/libp directory, 392
- /usr/lib/libw directory, 179
- /usr/lib/locale directory, 383, 392
- /usr/lib/lp directory, 392
- /usr/lib/lp directory, xiii
- /usr/lib/lp/lpsched command, 397
- /usr/lib/lpd command, 110, 227, 397
- /usr/lib/mail directory, 392
- /usr/lib/mail directory, xiii
- /usr/lib/netsvc directory, 239, 392
- /usr/lib/netsvc/yp/ypbind process, 253
- /usr/lib/netsvc/yp/ypbind process, 397
- /usr/lib/nfs directory, 392
- /usr/lib/nfs/lockd daemon, 397
- /usr/lib/nfs/mountd server, 397
- /usr/lib/nfs/nfsd daemon, 398
- /usr/lib/pics directory, 392
- /usr/lib/refer directory, 392
- /usr/lib/rsh command, 41
- /usr/lib/sa directory, 392
- /usr/lib/saf directory, 392
- /usr/lib/saf directory, xiii
- /usr/lib/spell directory, 393
- /usr/lib/uucp directory, 393
- /usr/lib/uucp directory, xiii
- /usr/local directory, 393
- /usr/net/servers directory, 393
- /usr/oasys directory, 393
- /usr/old directory, 87, 173, 393
- /usr/openwin directory, 393
- /usr/opt file, 75
- /usr/opt, 75
- /usr/sadm directory, 393
- /usr/sadm/bin directory, 393
- /usr/sadm/install directory, 393
- /usr/sbin directory, 82, 83, 87, 393, 397
- /usr/sbin/static directory, 393
- /usr/sccs directory, 168, 172, 173
- /usr/share directory, 393
- /usr/share file system mounting, 90
- /usr/share/lib directory, 393
- /usr/share/lib/keytables directory, 393
- /usr/share/lib/mailx directory, 393
- /usr/share/lib/nterm directory, 393
- /usr/share/lib/pub directory, 393
- /usr/share/lib/spell directory, 393
- /usr/share/lib/tabset directory, 393
- /usr/share/lib/terminfo database, 74, 110, 111, 385, 394, 397
- /usr/share/lib/tmac directory, 394
- /usr/share/lib/zoneinfo directory, 394
- /usr/share/man directory, 54
- /usr/share/man directory, 51
- /usr/share/src directory, 394
- /usr/snadm directory, 87, 394
- /usr/ucb directory, 36, 172, 394
- /usr/ucb/df command, 59, 79, 93, 213
- /usr/ucb/du command, 59, 60, 93, 215
- /usr/ucb/mail command, 50, 51, 207, 227
- /usr/ucbinclude directory, 394
- /usr/ucblib directory, 165, 167, 394
- /usr/vmsys directory, 394
- /usr/xpg2bin directory, 87
- /usr/xpg2include directory, 88
- /usr/xpg2lib directory, 88, 167
- /usr/xpg2lib/libxpg directory, 180
- /usr/xpg2lib/libxpg2.a library, 180
- ustar command, 250
- ustat system call , 281
- utime library routine, 366
- utimes system call , 281
- utmp system file, 386
- uuccheck command, 250
- uucico command, 250
- uucleanup command, 250
- UUCP (Unix-to-Unix Copy), 34, 119, 122
- uucp command, 238, 250

- auxiliary programs and daemons
 - directory, 393
- configuration information
 - directory, 389
- file deposit directory, 390
- log and status file directory, 121, 391
- queued jobs directory, 390
- /var/uucp directory, xiii
- /uucp directory, , *see* /etc/uucp directory
- uudecode command, 250
- uuencode command, 250
- uuencode system file, 386
- uuglist command, 120
- uulog command, 250
- uname command, 250
- uupick command, 251
- uusched command, 251
- uusend command, 251
- uustat command, 251
- uuto command, 251
- uux command, 251
- uuxqt command, 251

V

- vacation command, 251
- vadvise system call, 282
- val command, 173, 251
- valloc library routine, 366
- /var directory, 112, 389
- /var directory
 - changes in, 74, 88, 389, 391
 - described, 74, 76, 88, 389
- /var/adm directory, 389
- /var/crash directory, 389
- /var/cron directory, 389
- /var/lp directory, 390
- /var/lp directory, xiii
- /var/mail directory, 74, 88, 390, 397
- /var/mail directory, xiii
- /var/news directory, 390
- /var/nis directory, 390
- /var/opt directory, 390
- /var/opt/ directory, 88
- /var/options directory, 390
- /var/preserve directory, 390
- /var/sadm directory, 88, 390

- /var/saf directory, 88, 390
- /var/spool directory, 112, 390
- /var/spool/cron directory, 390
- /var/spool/locks directory, 390
- /var/spool/lp directory, 112, 390
- /var/spool/mail directory, 74, 88, 397
- /var/spool/mqueue directory, 390
- /var/spool/pkg directory, 390
- /var/spool/uucp directory, 390
- /var/spool/uucp directory, xiii
- /var/spool/uucppublic directory, 390
- /var/tmp directory, 390
- /var/uucp directory, 391
- /var/uucp/.Admin/account file, 121
- /var/uucp/.Admin/command file, 122
- /var/uucp/.Admin/perflog file, 121
- /var/uucp/.Admin/security file, 122
- /var/yp directory, 391
- varargs library routine, 366
- vax command, 251
- vc command, 173
- vedit command, 251
- version numbering, shared libraries, 153, 154, 164
- vfont system file, 386
- vfontinfo command, 251
- vfork system call , 282
- vfprintf library routine, 366
- VFS (virtual file-system architecture), 77, 82
- vfstab file, , *see* /etc/vfstab file\x0d
- vgrind command, 251
- vgrindefs database, 386
- vhangup system call, 282
- vi editor, 251, 390
- vidattr library routine, 366
- vidputs library routine, 366
- view command, 251
- vipw command, 251
- virtual file-system architecture (VFS), 77, 82
- vlimit library routine, 366
- vmstat command, 251
- /vmsys directory, 394
- /vmutex file, 18, 26, 67, 184, 186

- /vol file system, 76
- /vol file system, 75
- /vol/dev/aliases/cdrom0 file system, 62
- /vol/dev/aliases/floppy0 file system, 62
- volcancel command, 63
- volcheck command, 63
- volcopy command, 80, 96
- vold command, 63, 75
- vold.conf file, 63
- volmissing command, 63
- Volume Management
 - commands, 63
 - configuration files, 63
 - default file system for, 75
 - error messages, 63
 - overview, 24, 61
 - using, 61, 63
- vplot command, 251
- vprintf library routine, 367
- vsprintf library routine, 367
- vswap command, 251
- vsyslog library routine, 367
- vtimes library routine, 367
- vtroff command, 251
- vwidth command, 252
- vwprintw library routine, 367
- vwscanw library routine, 367

W

- w command, 252
- waddch library routine, 368
- waddstr library routine, 368
- wait command, 252
- wait system call , 282, 283
- wait3 system call , 282
- wait4 system call , 283
- waitpid system call, 283
- waitpid system call, 282
- wakeup command, 195
- wall command, 252
- wattroff library routine, 368
- wattron library routine, 368
- wattrset library routine, 369
- wc command, 252
- wclear library routine, 369
- wclrtoebot library routine, 369

- wclrtoeol library routine, 370
- wcstombs library routine, 370
- wctomb library routine, 370
- wdelch library routine, 370
- wdeleteln library routine, 371
- wechochar library routine, 371
- werase library routine, 371
- WEXITSTATUS system call , 257
- wgetch library routine, 371
- wgetstr library routine, 372
- what command, 173, 252
- whatis command, 252
- whatis database, 54, 204
- whereis command, 252
- which command, 252
- who command, 250, 252
- whoami command, 252
- whois command, 252
- WIFEXITED system call, 257
- WIFSIGNALED system call, 257
- WIFSTOPPED system call, 257
- WIFSTOPSIG system call, 257
- winch library routine, 372
- windex database, 54, 204
- window management services, 7
- window system, default, 48
- winsch library routine, 372
- winsertln library routine, 372
- wmove library routine, 373
- wnoutrefresh library routine, 373
- workspace manager, 7
- Workspace Programs submenu,
 - accessing, 134
- wprintw library routine, 373
- wrefresh library routine, 374
- write command, 252
- write system call, 284
- writev system call, 285
- wscanw library routine, 374
- wsetscrreg library routine, 374
- wstandend library routine, 375
- wstandout library routine, 375
- WTERMSIG system call, 257

X

xargs command, 252
xcrypt library routine, 375
xdecrypt library routine, 375
xdr_accepted_reply library routine, 375
xdr_array library routine, 375
xdr_authsys_parms library routine, 375
xdr_authunix_parms library routine, 375
xdr_bool library routine, 376
xdr_bytes library routine, 376
xdr_callhdr library routine, 376
xdr_callmsg library routine, 376
xdr_char library routine, 376
xdr_destroy library routine, 376
xdr_double library routine, 376
xdr_enum library routine, 376
xdr_float library routine, 376
xdr_free library routine, 376
xdr_getpos library routine, 376
xdr_inline library routine, 376
xdr_int library routine, 376
xdr_long library routine, 376
xdr_opaque library routine, 376
xdr_pointer library routine, 376
xdr_reference library routine, 376
xdr_setpos library routine, 376
xdr_short library routine, 376
xdr_string library routine, 376
xdr_u_char library routine, 376
xdr_u_int library routine, 376
xdr_u_long library routine, 377
xdr_u_short library routine, 377
xdr_union library routine, 377
xdr_vector library routine, 377
xdr_void library routine, 377
xdr_wrapstring library routine, 377
xdrmem_create library routine, 377
xdrrec_create library routine, 377
xdrrec_endofrecord library routine, 377
xdrrec_eof library routine, 377
xdrrec_skiprecord library routine, 377
xdrstdio_create library routine, 377
XENIX semaphore file system
 (xnamefs), 79
XENIX, Solaris SPARC release and, 10
xget command, 252

xgettext command, 180
xlock command (OpenWindows), 226
xnamefs, 79
 /xpg2bin directory, 87
 /xpg2include directory, 88
 /xpg2lib directory, , see /usr/xpg2lib
 directory\x0d
xsend command, 252
xset command (OpenWindows), 241
xstr command, 252
xtab system file, 386, 397
xtom library routine, 377
xv_get_sel command, 219
XView File Chooser, OpenWindows
 Developer's Guide File
 Chooser vs., 49
XView Window Toolkit, 171

Y

y0 library routine, 378
y1 library routine, 379
yacc command, 173, 252
yaccpar command, 173
yes command, 253
yn library routine, 379
yp database directory, 391
ypaliases system file, 386
ypbatchupd command, 253
ypbind command, 253, 397
ypcat command, 253
ypfiles database, 386
ypgroup system file, 386
ypinit command, 253
ypmatch command, 253
yppasswd command, 253
yppasswd system file, 386
yppoll command, 253
ypprintcap database, 386
yppush command, 253
ypserv command, 253
ypset command, 253
ypupdated command, 253
ypwhich command, 253
ypxfr command, 253
ypxfrd command, 253

Z

zcat command, 253
zdump command, 254

zic command, 254
/zoneinfo directory, 394