

SunOS Reference Manual

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



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

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

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

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and FAR 52.227-19.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

TRADEMARKS

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

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

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

X Window System is a trademark of the X Consortium.

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

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

Portions © AT&T 1983-1990 and reproduced with permission from AT&T.

Preface

OVERVIEW

A man page is provided for both the naive user, and sophisticated user who is familiar with the SunOS operating system and is in need of on-line information. A man page is intended to answer concisely the question “What does it do?” The man pages in general comprise a reference manual. They are not intended to be a tutorial.

The following contains a brief description of each section in the man pages and the information it references:

- Section 1 describes, in alphabetical order, commands available with the operating system.
- Section 1M describes, in alphabetical order, commands that are used chiefly for system maintenance and administration purposes.
- Section 2 describes all of the system calls. Most of these calls have one or more error returns. An error condition is indicated by an otherwise impossible returned value.
- Section 3 describes functions found in various libraries, other than those functions that directly invoke UNIX system primitives, which are described in Section 2 of this volume.

-
- Section 4 outlines the formats of various files. The C structure declarations for the file formats are given where applicable.
 - Section 5 contains miscellaneous documentation such as character set tables, etc.
 - Section 6 contains available games and demos.
 - Section 7 describes various special files that refer to specific hardware peripherals, and device drivers. STREAMS software drivers, modules and the STREAMS-generic set of system calls are also described.
 - Section 9 provides reference information needed to write device drivers in the kernel operating systems environment. It describes two device driver interface specifications: the Device Driver Interface (DDI) and the Driver–Kernel Interface (DKI).
 - Section 9E describes the DDI/DKI, DDI-only, and DKI-only entry-point routines a developer may include in a device driver.
 - Section 9F describes the kernel functions available for use by device drivers.
 - Section 9S describes the data structures used by drivers to share information between the driver and the kernel.

Below is a generic format for man pages. The man pages of each manual section generally follow this order, but include only needed headings. For example, if there are no bugs to report, there is no BUGS section. See the intro pages for more information and detail about each section, and **man(1)** for more information about man pages in general.

NAME

This section gives the names of the commands or functions documented, followed by a brief description of what they do.

SYNOPSIS

This section shows the syntax of commands or functions. When a command or file does not exist in the standard path, its full pathname is shown. Literal characters (commands and options) are in **bold** font and variables (arguments, parameters and substitution characters) are in *italic* font. Options and

arguments are alphabetized, with single letter arguments first, and options with arguments next, unless a different argument order is required.

The following special characters are used in this section:

- [] The option or argument enclosed in these brackets is optional. If the brackets are omitted, the argument *must* be specified.
- ... Ellipses. Several values may be provided for the previous argument, or the previous argument can be specified multiple times, for example, '*filename ...*'.
- | Separator. Only one of the arguments separated by this character can be specified at time.
- { } Braces. The options and/or arguments enclosed within braces are interdependent, such that everything enclosed must be treated as a unit.

PROTOCOL

This section occurs only in subsection 3R to indicate the protocol description file. The protocol specification pathname is always listed in **bold** font.

AVAILABILITY

This section briefly states any limitations on the availability of the command. These limitations could be hardware or software specific.

A specification of a class of hardware platform, such as **x86** or **SPARC**, denotes that the command or interface is applicable for the hardware platform specified.

In Section 1 and Section 1M, **AVAILABILITY** indicates which package contains the command being described on the manual page. In order to use the command, the specified package must have been installed with the operating system. If the package was not installed, see **pkgadd(1)** for information on how to upgrade.

MT-LEVEL

This section lists the **MT-LEVEL** of the library functions described in the Section 3 manual pages. The **MT-LEVEL** defines the libraries' ability to support threads. See **Intro(3)** for more information.

DESCRIPTION

This section defines the functionality and behavior of the service. Thus it describes concisely what the command does. It does not discuss *OPTIONS* or cite *EXAMPLES*. Interactive commands, subcommands, requests, macros, functions and such, are described under *USAGE*.

IOCTL

This section appears on pages in Section 7 only. Only the device class which supplies appropriate parameters to the **ioctl(2)** system call is called **ioctl** and generates its own heading. **ioctl** calls for a specific device are listed alphabetically (on the man page for that specific device). **ioctl** calls are used for a particular class of devices all of which have an **io** ending, such as **mtio(7)**.

OPTIONS

This lists the command options with a concise summary of what each option does. The options are listed literally and in the order they appear in the *SYNOPSIS* section. Possible arguments to options are discussed under the option, and where appropriate, default values are supplied.

OPERANDS

This section lists the command operands and describes how they affect the actions of the command.

OUTPUT

This section describes the output - standard output, standard error, or output files - generated by the command.

RETURN VALUES

If the man page documents functions that return values, this section lists these values and describes the conditions under which they are returned. If a function can return only constant values, such as 0 or -1, these values are listed in tagged paragraphs. Otherwise, a single paragraph describes the return values of each function. Functions declared as **void** do not return values, so they are not discussed in *RETURN VALUES*.

ERRORS

On failure, most functions place an error code in the global variable **errno** indicating why they failed. This section lists alphabetically all error codes a function can generate and describes the conditions that cause each error. When more than one condition can cause the same error, each condition is described in a separate paragraph under the error code.

USAGE

This section is provided as a *guidance* on use. This section lists special rules, features and commands that require in-depth explanations. The subsections listed below are used to explain built-in functionality:

- Commands**
- Modifiers**
- Variables**
- Expressions**
- Input Grammar**

EXAMPLES

This section provides examples of usage or of how to use a command or function. Wherever possible a complete example including command line entry and machine response is shown. Whenever an example is given, the prompt is shown as

example%

or if the user must be super-user,

example#

Examples are followed by explanations, variable substitution rules, or returned values. Most examples illustrate concepts from the SYNOPSIS, DESCRIPTION, OPTIONS and USAGE sections.

ENVIRONMENT

This section lists any environment variables that the command or function affects, followed by a brief description of the effect.

EXIT STATUS

This section lists the values the command returns to the calling program or shell and the conditions that cause these values to be returned. Usually, zero is returned for successful completion and values greater than zero for various error conditions.

FILES

This section lists all filenames referred to by the man page, files of interest, and files created or required by commands. Each is followed by a descriptive summary or explanation.

SEE ALSO

This section lists references to other man pages, in-house documentation and outside publications.

DIAGNOSTICS

This section lists diagnostic messages with a brief explanation of the condition causing the error. Messages appear in **bold** font with the exception of variables, which are in *italic* font.

WARNINGS

This section lists warnings about special conditions which could seriously affect your working conditions — this is not a list of diagnostics.

NOTES

This section lists additional information that does not belong anywhere else on the page. It takes the form of an *aside* to the user, covering points of special interest. Critical information is never covered here.

BUGS

This section describes known bugs and wherever possible suggests workarounds.

NAME	Intro, intro – introduction to maintenance commands and application programs
AVAILABILITY	This section indicates which package contains the commands being described on this page. To be able to use the command, the indicated package must have been installed with the operating system. For information on how to add a package see pkgadd(1) .
DESCRIPTION	<p>This section describes, in alphabetical order, commands that are used chiefly for system maintenance and administration purposes.</p> <p>Because of command restructuring for the Virtual File System architecture, there are several instances of multiple manual pages that begin with the same name. For example, the mount, pages – mount(1M), mount_cachefs(1M), mount_hfs(1M), mount_nfs(1M), mount_tmpfs(1M), and mount_ufs(1M). In each such case the first of the multiple pages describes the syntax and options of the generic command, that is, those options applicable to all FSTypes (file system types). The succeeding pages describe the functionality of the FSType-specific modules of the command. These pages list the command followed by an underscore (_) and the FSType to which they pertain. Note that the administrator should not attempt to call these modules directly. The generic command provides a common interface to all of them. Thus the FSType-specific manual pages should not be viewed as describing distinct commands, but rather as detailing those aspects of a command that are specific to a particular FSType.</p>
COMMAND SYNTAX	<p>Unless otherwise noted, commands described in this section accept options and other arguments according to the following syntax:</p> <p style="text-align: center;"><i>name</i> [<i>option(s)</i>] [<i>cmdarg(s)</i>]</p> <p>where:</p> <p><i>name</i> The name of an executable file</p> <p><i>option</i> – <i>noargletter(s)</i> or, – <i>argletter</i>< ><i>optarg</i> where < > is optional white space</p> <p><i>noargletter</i> A single letter representing an option without an argument</p> <p><i>argletter</i> A single letter representing an option requiring an argument</p> <p><i>optarg</i> Argument (character string) satisfying preceding <i>argletter</i></p> <p><i>cmdarg</i> Pathname (or other command argument) <i>not</i> beginning with – or, – by itself indicating the standard input</p>
SEE ALSO	getopt(1) , pkgadd(1M) , getopt(3C)
DIAGNOSTICS	Upon termination, each command returns 0 for normal termination and non-zero to indicate troubles such as erroneous parameters, bad or inaccessible data, or other inability to cope with the task at hand. It is called variously “exit code,” “exit status,” or “return code,” and is described only where special conventions are involved.

NOTES

Unfortunately, not all commands adhere to the standard syntax.

Name	Description
accept (1M)	accept or reject print requests
acct (1M)	overview of accounting and miscellaneous accounting commands
acctcms (1M)	command summary from process accounting records
acctcon (1M)	connect-time accounting
acctcon1 (1M)	See acctcon (1M)
acctcon2 (1M)	See acctcon (1M)
acctdisk (1M)	See acct (1M)
acctdusg (1M)	See acct (1M)
acctmerg (1M)	merge or add total accounting files
accton (1M)	See acct (1M)
acctprc (1M)	process accounting
acctprc1 (1M)	See acctprc (1M)
acctprc2 (1M)	See acctprc (1M)
acctsh (1M)	shell procedures for accounting
acctwtmp (1M)	See acct (1M)
adbgen (1M)	generate adb script
addbadsec (1M)	map out defective disk blocks
add_drv (1M)	add a new device driver to the system
add_install_client (1M)	See install_scripts (1M)
admintool (1M)	system administration with a graphical user interface
aliasadm (1M)	manipulate the NIS+ aliases map
allocate (1M)	device allocation
arp (1M)	address resolution display and control
aset (1M)	monitors or restricts accesses to system files and directories
aset.restore (1M)	restores system files to their content before ASET is installed
aspppd (1M)	Asynchronous PPP Link Manager
aspppls (1M)	See aspppd (1M)
audit (1M)	control the behavior of the audit daemon
auditconfig (1M)	configure auditing

auditd (1M)	audit daemon
auditreduce (1M)	merge and select audit records from audit trail files
audit_startup (1M)	audit subsystem initialization script
auditstat (1M)	display kernel audit statistics
audit_warn (1M)	audit daemon warning script
automount (1M)	install automatic mount points
automountd (1M)	autofs mount/unmount daemon
autopush (1M)	configures lists of automatically pushed STREAMS modules
bdconfig (1M)	configures the bd (buttons and dials) stream
boot (1M)	start the system kernel or a standalone program
bootparamd (1M)	See rpc.bootparamd (1M)
bsmconv (1M)	enable/disable the Basic Security Module (BSM) on Solaris
bsmunconv (1M)	See bsmconv (1M)
cachefslog (1M)	Cache File System logging
cachefsstat (1M)	Cache File System statistics
cachefswssize (1M)	Determine working set size for cachefs
captoinfo (1M)	convert a termcap description into a terminfo description
catman (1M)	create the cat files for the manual
cfsadmin (1M)	administer disk space used for caching file systems with the Cache File-System (CacheFS)
cg14config (1M)	configure the cgfourteen device
chargefee (1M)	See acctsh (1M)
check (1M)	See install_scripts (1M)
chown (1M)	change owner
chroot (1M)	change root directory for a command
chrtbl (1M)	generate character classification and conversion tables
ckpacct (1M)	See acctsh (1M)
closewtmp (1M)	See acct (1M)
cli (1M)	clear inode
colltbl (1M)	create string collation routines
comsat (1M)	See in.comsat (1M)
convert4x (1M)	converts SunOS 4.x clients to work from Solaris 2.x servers
crash (1M)	examine system images

cron (1M)	clock daemon
dcopy (1M)	See clri (1M)
dd (1M)	convert and copy a file
deallocate (1M)	device deallocation
devattr (1M)	lists device attributes
devconfig (1M)	configure device attributes
devfree (1M)	release devices from exclusive use
devinfo (1M)	print device specific information
devlinks (1M)	adds /dev entries for miscellaneous devices and pseudo-devices
devnm (1M)	device name
devreserv (1M)	reserve devices for exclusive use
df (1M)	report number of free disk blocks and files
dfmounts (1M)	display mounted resource information
dfmounts_nfs (1M)	display mounted NFS resource information
dfshares (1M)	list available resources from remote or local systems
dfshares_nfs (1M)	list available NFS resources from remote systems
df_ufs (1M)	report free disk space on ufs file systems
discover4x (1M)	analyze the SunOS 4.x client support after converting to Solaris 2.x
disks (1M)	adds /dev entries for hard disks attached to the system
diskscan (1M)	perform surface analysis
dispadmin (1M)	process scheduler administration
dmesg (1M)	collect system diagnostic messages to form error log
dminfo (1M)	report information about a device entry in a device maps file
dodisk (1M)	See acctsh (1M)
domainname (1M)	set or display name of the current domain
drvconfig (1M)	configure the /devices directory
du (1M)	summarize disk usage
edquota (1M)	edit user quotas for ufs file system
eeeprom (1M)	EEPROM display and load utility
fdetach (1M)	detach a name from a STREAMS-based file descriptor
fdisk (1M)	create or modify fixed disk partition table
ff (1M)	list file names and statistics for a file system
ff_ufs (1M)	list file names and statistics for a ufs file system

fingerd (1M)	See in.fingerd (1M)
firmware (1M)	bootable firmware programs and firmware commands
fmthard (1M)	populate VTOC on hard disks
fncheck (1M)	check for consistency between FNS data and NIS+ data
fncreate (1M)	Create an FNS context
fncreate_fs (1M)	create FNS file system contexts
fncreate_printer (1M)	create new printers in the FNS namespace
fndestroy (1M)	Destroy an FNS context
format (1M)	disk partitioning and maintenance utility
fsck (1M)	check and repair file systems
fsck_cacheofs (1M)	check integrity of data cached with CacheFS
fsck_s5fs (1M)	file system consistency check and interactive repair
fsck_ufs (1M)	file system consistency check and interactive repair
fsdb (1M)	file system debugger
fsdb_ufs (1M)	ufs file system debugger
fsirand (1M)	install random inode generation numbers
fstyp (1M)	determine file system type
ftpd (1M)	See in.ftpd (1M)
fusage (1M)	disk access profiler
fuser (1M)	identify processes using a file or file structure
fwtmp (1M)	manipulate connect accounting records
gencc (1M)	create a front-end to the cc command
getdev (1M)	lists devices based on criteria
getdgrp (1M)	lists device groups which contain devices that match criteria
getent (1M)	get entries from administrative database
gettable (1M)	get DoD Internet format host table from a host
getty (1M)	set terminal type, modes, speed, and line discipline
getvol (1M)	verifies device accessibility
groupadd (1M)	add (create) a new group definition on the system
groupdel (1M)	delete a group definition from the system
groupmod (1M)	modify a group definition on the system
grpck (1M)	See pwck (1M)
halt (1M)	stop the processor
hostconfig (1M)	configure a system's host parameters

htable(1M)	convert DoD Internet format host table
id(1M)	return user identity
ifconfig(1M)	configure network interface parameters
in.comsat(1M)	biff server
inetd(1M)	Internet services daemon
in.fingerd(1M)	remote user information server
infocmp(1M)	compare or print out terminfo descriptions
in.ftpd(1M)	file transfer protocol server
init(1M)	process control initialization
in.named(1M)	Internet domain name server
in.rarpd(1M)	DARPA Reverse Address Resolution Protocol server
in.rdisc(1M)	network router discovery daemon
in.rexecd(1M)	remote execution server
in.rlogind(1M)	remote login server
in.routed(1M)	network routing daemon
in.rshd(1M)	remote shell server
in.rwhod(1M)	system status server
install(1M)	install commands
install4x(1M)	install SunOS 4.x diskless/dataless client support on Solaris 2.x servers
installboot(1M)	install bootblocks in a disk partition
installf(1M)	add a file to the software installation database
install_scripts(1M)	scripts used to install the Solaris software
in.talkd(1M)	server for talk program
in.telnetd(1M)	DARPA TELNET protocol server
in.tftpd(1M)	Internet Trivial File Transfer Protocol server
in.tnamed(1M)	DARPA trivial name server
in.uucpd(1M)	UUCP server
iostat(1M)	report I/O statistics
kadb(1M)	a kernel debugger
kdmconfig(1M)	configure or unconfigure keyboard, display, and mouse options for OpenWindows and internationalization
kerbd(1M)	generates and validates Kerberos tickets for kernel RPC
kernel(1M)	UNIX system executable file containing basic operating system services
keyserv(1M)	server for storing private encryption keys

killall (1M)	kill all active processes
labelit (1M)	list or provide labels for file systems
labelit_hfs (1M)	provide and print labels for hfs file systems
labelit_ufs (1M)	provide and print labels for ufs file systems
lastlogin (1M)	See acctsh (1M)
leoconfig (1M)	initialize the Leo (ZX and TZX) Graphics Accelerator and download microcode
link (1M)	link and unlink files and directories
list_devices (1M)	list allocatable devices
listdgrp (1M)	lists members of a device group
listen (1M)	network listener daemon
lockd (1M)	network lock daemon
lockfs (1M)	change or report file system locks
logins (1M)	list user and system login information
lpadmin (1M)	configure the LP print service
lpfilter (1M)	administer filters used with the LP print service
lpforms (1M)	administer forms used with the LP print service
lpmove (1M)	See lpsched (1M)
lpsched (1M)	start/stop the LP print service and move requests
lpshut (1M)	See lpsched (1M)
lpssystem (1M)	register remote systems with the print service
lpusers (1M)	set printing queue priorities
mail.local (1M)	store mail in a mailbox
makedbm (1M)	make a dbm file, or get a text file from a dbm file
mk (1M)	remake the binary system and commands from source code
mkfifo (1M)	make FIFO special file
mkfile (1M)	create a file
mkfs (1M)	construct a file system
mkfs_ufs (1M)	construct a ufs file system
mknod (1M)	make a special file
modinfo (1M)	display information about loaded kernel modules
modload (1M)	load a kernel module
modunload (1M)	unload a module
monacct (1M)	See acctsh (1M)
monitor (1M)	SPARC system PROM monitor

montbl (1M)	create monetary database
mount (1M)	mount or unmount file systems and remote resources
mountall (1M)	mount, unmount multiple file systems
mount_cacheufs (1M)	mount CacheFS file systems
mountd (1M)	NFS mount request server
mount_hfs (1M)	mount hfs file systems
mount_nfs (1M)	mount remote NFS resources
mount_pcfs (1M)	mount pcfs file systems
mount_s5fs (1M)	mount s5 file systems
mount_tmpfs (1M)	mount tmpfs file systems
mount_ufs (1M)	mount ufs file systems
mpstat (1M)	report per-processor statistics
mvd (1M)	move a directory
named (1M)	See in.named (1M)
named-xfer (1M)	See in.named (1M)
ncheck (1M)	generate a list of path names versus i-numbers
ncheck_ufs (1M)	generate pathnames versus i-numbers for ufs file systems
ndd (1M)	get and set driver configuration parameters
netstat (1M)	show network status
newfs (1M)	construct a new UFS file system
newkey (1M)	create a new Diffie-Hellman key pair in the publickey database
nfsd (1M)	NFS daemon
nfsstat (1M)	NFS statistics
nisaddcred (1M)	create NIS+ credentials
nisaddent (1M)	create NIS+ tables from corresponding /etc files or NIS maps
nis_cachemgr (1M)	NIS+ utility to cache location information about NIS+ servers
nisclient (1M)	initialize NIS+ credentials for NIS+ principals
nisd (1M)	See rpc.nisd (1M)
nisd_resolv (1M)	See rpc.nisd_resolv (1M)
nisinit (1M)	NIS+ client and server initialization utility
nislog (1M)	display the contents of the NIS+ transaction log
nispawdd (1M)	See rpc.nispawdd (1M)
nisping (1M)	send ping to NIS+ servers

nispopulate (1M)	populate the NIS+ tables in a NIS+ domain.
nisserver (1M)	set up NIS+ servers.
nissetup (1M)	initialize a NIS+ domain
nisshowcache (1M)	NIS+ utility to print out the contents of the shared cache file
nisstat (1M)	report NIS+ server statistics
nisupdkeys (1M)	update the public keys in a NIS+ directory object
nlsadmin (1M)	network listener service administration
nscd (1M)	name service cache daemon
nslookup (1M)	query name servers interactively
nstest (1M)	DNS test shell
nulladm (1M)	See acctsh (1M)
obpsym (1M)	Kernel Symbolic Debugging for OpenBoot Firmware
passmgmt (1M)	password files management
pbind (1M)	control and query bindings of processes to processors
pcmcia (1M)	PCMCIA user daemon
pfinstall (1M)	tests installation profiles
ping (1M)	send ICMP ECHO_REQUEST packets to network hosts
pkgadd (1M)	transfer software package to the system
pkgask (1M)	stores answers to a request script
pkgchk (1M)	check accuracy of installation
pkgrm (1M)	removes a package from the system
pmadm (1M)	port monitor administration
ports (1M)	create /dev entries and inittab entries for serial lines
poweroff (1M)	See halt (1M)
praudit (1M)	print contents of an audit trail file
prctmp (1M)	See acctsh (1M)
prdaily (1M)	See acctsh (1M)
prtacct (1M)	See acctsh (1M)
prtconf (1M)	print system configuration
prtdiag (1M)	print system diagnostic information
prtvtoc (1M)	report information about a disk geometry and partitioning
psradm (1M)	set processors online or offline
psrinfo (1M)	print processor information
putdev (1M)	edits device table

putdgrp(1M)	edits device group table
pwck(1M)	password/group file checkers
pwconv(1M)	installs and updates /etc/shadow with information from /etc/passwd
quot(1M)	summarize file system ownership
quota(1M)	display a user's ufs file system disk quota and usage
quotacheck(1M)	ufs file system quota consistency checker
quotaoff(1M)	See quotaon(1M)
quotaon(1M)	turn ufs file system quotas on and off
rarpd(1M)	See in.rarpd(1M)
rdate(1M)	set system date from a remote host
rdisc(1M)	See in.rdisc(1M)
reboot(1M)	restart the operating system
reject(1M)	See accept(1M)
rem_drv(1M)	remove a device driver from the system
removef(1M)	remove a file from software database
repquota(1M)	summarize quotas for a ufs file system
re-preinstall(1M)	installs the JumpStart software on a system
restricted_shell(1M)	See rsh(1M)
rexd(1M)	See rpc.rexd(1M)
rexecd(1M)	See in.rexecd(1M)
rlogind(1M)	See in.rlogind(1M)
rm_install_client(1M)	See install_scripts(1M)
rmmount(1M)	removable media mounter that automatically mounts a file system on a CD-ROM and floppy
rmt(1M)	remote magtape protocol module
route(1M)	manually manipulate the routing tables
routed(1M)	See in.routed(1M)
rpcbind(1M)	universal addresses to RPC program number mapper
rpc.bootparamd(1M)	boot parameter server
rpcinfo(1M)	report RPC information
rpc.nisd(1M)	NIS+ service daemon
rpc.nisd_resolv(1M)	NIS+ service daemon
rpc.nispasswd(1M)	NIS+ password update daemon
rpc.rexd(1M)	RPC-based remote execution server
rpc.rstatd(1M)	kernel statistics server

rpc.rusersd (1M)	network username server
rpc.rwalld (1M)	network rwall server
rpc.sprayd (1M)	spray server
rpld (1M)	x86 Network Booting RPL (Remote Program Load) Server
rquotad (1M)	remote quota server
rsh (1M)	restricted shell command interpreter
rshd (1M)	See in.rshd (1M)
rstatd (1M)	See rpc.rstatd (1M)
rtc (1M)	provide all real-time clock and GMT-lag management
runacct (1M)	run daily accounting
runacct (1M)	See acctsh (1M)
rusersd (1M)	See rpc.rusersd (1M)
rwall (1M)	write to all users over a network
rwalld (1M)	See rpc.rwalld (1M)
rwhod (1M)	See in.rwhod (1M)
sa1 (1M)	See sar (1M)
sa2 (1M)	See sar (1M)
sac (1M)	service access controller
sacadm (1M)	service access controller administration
sadc (1M)	See sar (1M)
sadmin (1M)	distributed system administration daemon
saf (1M)	Service Access Facility
sar (1M)	system activity report package
savecore (1M)	save a core dump of the operating system
sendmail (1M)	send mail over the internet
server_upgrade (1M)	upgrade clients of a heterogeneous OS server
setmnt (1M)	establish mount table
setuname (1M)	changes machine information
setup_install_server (1M)	See install_scripts (1M)
share (1M)	make local resource available for mounting by remote systems
shareall (1M)	share, unshare multiple resources
share_nfs (1M)	make local NFS filesystems available for mounting by remote systems
showmount (1M)	show all remote mounts

showrev (1M)	show machine and software revision information
shutacct (1M)	See acctsh (1M)
shutdown (1M)	shut down system, change system state
snoop (1M)	capture and inspect network packets
spray (1M)	spray packets
sprayd (1M)	See rpc.sprayd (1M)
ssaadm (1M)	Administration program for the SPARCstorage Array
ssacli (1M)	SPARCstorage Array Command Line Interface
startup (1M)	See acctsh (1M)
statd (1M)	network status monitor
strace (1M)	print STREAMS trace messages
strclean (1M)	STREAMS error logger cleanup program
strerr (1M)	STREAMS error logger daemon
sttydefs (1M)	maintain line settings and hunt sequences for TTY ports
su (1M)	become super-user or another user
sulogin (1M)	access single-user mode
suninstall (1M)	install the Solaris environment
swap (1M)	swap administrative interface
swmtool (1M)	install, upgrade, and remove software packages
sxconfig (1M)	configure contiguous memory for the SX video subsystem
sync (1M)	update the super block
syncinit (1M)	set serial line interface operating parameters.
syncloop (1M)	synchronous serial loopback test program
syncstat (1M)	report driver statistics from a synchronous serial link.
sysdef (1M)	output system definition
sysidconfig (1M)	execute system configuration applications, or define set of system configuration applications
sysidnet (1M)	See sysidtool (1M)
sysidnis (1M)	See sysidtool (1M)
sysidroot (1M)	See sysidtool (1M)
sysidsys (1M)	See sysidtool (1M)
sysidtool (1M)	system configuration
syslogd (1M)	log system messages
sys-unconfig (1M)	undo a system's configuration
talkd (1M)	See in.talkd (1M)

tapes(1M)	adds /dev entries for tape drives attached to the system
taskstat(1M)	prints ASET tasks status
tcxconfig(1M)	configure the default linearity of the 24-bit TrueColor Visual for OpenWindows on a system with an S24 frame buffer (TCX)
telinit(1M)	See init(1M)
telnetd(1M)	See in.telnetd(1M)
tftpd(1M)	See in.tftpd(1M)
tic(1M)	terminfo compiler
tnamed(1M)	See in.tnamed(1M)
ttyadm(1M)	format and output port monitor-specific information
tyhstmgr(1M)	administer hosts on a network
ttymon(1M)	port monitor for terminal ports
tunefs(1M)	tune up an existing file system
turnacct(1M)	See acctsh(1M)
uadmin(1M)	administrative control
ufsdump(1M)	incremental file system dump
ufsrestore(1M)	incremental file system restore
umount(1M)	See mount(1M)
umountall(1M)	See mountall(1M)
unlink(1M)	See link(1M)
unshare(1M)	make local resource unavailable for mounting by remote systems
unshareall(1M)	See shareall(1M)
unshare_nfs(1M)	make local NFS filesystems unavailable for mounting by remote systems
useradd(1M)	administer a new user login on the system
userdel(1M)	delete a user's login from the system
usermod(1M)	modify a user's login information on the system
utmp2wtmp(1M)	See acct(1M)
uuccheck(1M)	check the uucp directories and permissions file
uucico(1M)	file transport program for the uucp system
uucleanup(1M)	uucp spool directory clean-up
uucpd(1M)	See in.uucpd(1M)
uusched(1M)	the scheduler for the uucp file transport program
uutry(1M)	See Uutry(1M)

Utry (1M)	try to contact remote system with debugging on
uuxqt (1M)	execute remote command requests
vmstat (1M)	report virtual memory statistics
volcopy (1M)	make an image copy of file system
volcopy_ufs (1M)	make an image copy of a ufs file system
vold (1M)	Volume Management daemon to manage CD-ROM and floppy devices
wall (1M)	write to all users
wchrtbl (1M)	See chrtbl (1M)
whodo (1M)	who is doing what
wtmpfix (1M)	See fwtmp (1M)
ypbind (1M)	NIS binder process
ypinit (1M)	set up NIS client
ypmake (1M)	rebuild NIS database
yppoll (1M)	return current version of a NIS map at a NIS server host
ypset (1M)	point ypbind at a particular server
ypxfr (1M)	transfer NIS map from a NIS server to host
zdump (1M)	time zone dumper
zic (1M)	time zone compiler

NAME	Uutry, uutry – try to contact remote system with debugging on
SYNOPSIS	<i>/usr/lib/uucp/Uutry</i> [<i>-r</i>] [<i>-ctype</i>] [<i>-xdebug-level</i>] <i>system-name</i>
DESCRIPTION	Uutry is a shell script that is used to invoke uucico (1M) to call a remote site. Debugging is initially turned on and is set to the default value of 5 . The debugging output is put in file <i>/tmp/system-name</i> .
OPTIONS	<p><i>-r</i> This option overrides the retry time that is set in file <i>/var/uucp/.Status/system-name</i>.</p> <p><i>-ctype</i> The first field in the Devices file is the "Type" field. The <i>-c</i> option forces uucico to only use entries in the "Type" field that match the user specified <i>type</i>. The specified <i>type</i> is usually the name of a local area network.</p> <p><i>-xdebug-level</i> <i>debug-level</i> is a number from 0 to 9. Higher numbers give more detailed debugging information.</p>
FILES	<p><i>/etc/uucp/Devices</i> <i>/etc/uucp/Limits</i> <i>/etc/uucp/Permissions</i> <i>/etc/uucp/Systems</i> <i>/tmp/system-name</i> <i>/var/spool/locks/*</i> <i>/var/spool/uucp/*</i> <i>/var/spool/uucppublic/*</i></p>
SEE ALSO	uucp (1C), uux (1C), uucico (1M)

NAME	accept, reject – accept or reject print requests
SYNOPSIS	accept <i>destinations</i> reject [-r <i>reason</i>] <i>destinations</i>
AVAILABILITY	SUNWlpu
DESCRIPTION	accept allows the queueing of print requests for the named <i>destinations</i> . A <i>destination</i> can be either a printer or a class of printers. Run lpstat -a to find the status of <i>destinations</i> . reject prevents queueing of print requests for the named <i>destinations</i> . A <i>destination</i> can be either a printer or a class of printers. Run lpstat -a to find the status of <i>destinations</i> .
OPTIONS	The following option is useful with reject . -r <i>reason</i> Assign a <i>reason</i> for rejection of requests. This <i>reason</i> applies to all <i>destinations</i> specified. <i>reason</i> is reported by lpstat -a . It must be enclosed in quotes if it contains blanks. The default reason is unknown reason for existing destinations, and new destination for destinations just added to the system but not yet accepting requests.
FILES	/var/spool/lp/*
SEE ALSO	enable(1) , lp(1) , lpstat(1) , lpadm(1M) , lpsched(1M)

NAME	acct, acctdisk, acctdusg, accton, acctwtmp, closewtmp, utmp2wtmp – overview of accounting and miscellaneous accounting commands
SYNOPSIS	<pre> /usr/lib/acct/acctdisk /usr/lib/acct/acctdusg [-ufilename] [-pfilename] /usr/lib/acct/accton [filename] /usr/lib/acct/acctwtmp reason /usr/lib/acct/closewtmp /usr/lib/acct/utmp2wtmp </pre>
AVAILABILITY	SUNWaccu
DESCRIPTION	<p>Accounting software is structured as a set of tools (consisting of both C programs and shell procedures) that can be used to build accounting systems. acctsh(1M) describes the set of shell procedures built on top of the C programs.</p> <p>Connect time accounting is handled by various programs that write records into <i>/var/adm/wtmp</i>, as described in utmp(4). The programs described in acctcon(1M) convert this file into session and charging records, which are then summarized by acctmerg(1M).</p> <p>Process accounting is performed by the system kernel. Upon termination of a process, one record per process is written to a file (normally <i>/var/adm/pacct</i>). The programs in acctprc(1M) summarize this data for charging purposes; acctcms(1M) is used to summarize command usage. Current process data may be examined using acctcom(1).</p> <p>Process accounting records and connect time accounting records (or any accounting records in the tacct format described in acct(4)) can be merged and summarized into total accounting records by acctmerg (see tacct format in acct(4)). prtacct (see acctsh(1M)) is used to format any or all accounting records.</p> <p>acctdisk reads lines that contain user ID, login name, and number of disk blocks and converts them to total accounting records that can be merged with other accounting records.</p> <p>acctdusg reads its standard input (usually from find / -print) and computes disk resource consumption (including indirect blocks) by login.</p> <p>accton without arguments turns process accounting off. If <i>filename</i> is given, it must be the name of an existing file, to which the kernel appends process accounting records (see acct(2) and acct(4)).</p> <p>acctwtmp writes a utmp(4) record to its standard output. The record contains the current time and a string of characters that describe the <i>reason</i>. A record type of ACCOUNTING is assigned (see utmp(4)). <i>reason</i> must be a string of 11 or fewer characters, numbers, \$, or spaces. For example, the following are suggestions for use in reboot and shutdown procedures, respectively:</p>

```
acctwtmp "acctg on" >> /var/adm/wtmp
acctwtmp "acctg off" >> /var/adm/wtmp
```

For each user currently logged on, **closewtmp** puts a false **DEAD_PROCESS** record in the **/var/adm/wtmp** file. **runacct** (see **runacct(1M)**) uses this false **DEAD_PROCESS** record so that the connect accounting procedures can track the time used by users logged on before **runacct** was invoked.

For each user currently logged on, **runacct** uses **utmp2wtmp** to create an entry in the file **/var/adm/wtmp**, created by **runacct**. Entries in **/var/adm/wtmp** enable subsequent invocations of **runacct** to account for connect times of users currently logged in.

OPTIONS

-ufilename Records consisting of those filenames for which **acctdusg** charges no one are placed in *filename* (a potential source for finding users trying to avoid disk charges).

-pfilename Specify a password file, *filename*. This option is not needed if the password file is **/etc/passwd**.

ENVIRONMENT

If any of the **LC_*** variables (**LC_CTYPE**, **LC_MESSAGES**, **LC_TIME**, **LC_COLLATE**, **LC_NUMERIC**, and **LC_MONETARY**) (see **environ(5)**) are not set in the environment, the operational behavior of **acct** for each corresponding locale category is determined by the value of the **LANG** environment variable. If **LC_ALL** is set, its contents are used to override both the **LANG** and the other **LC_*** variables. If none of the above variables are set in the environment, the "C" (U.S. style) locale determines how **acct** behaves.

LC_CTYPE determines how **acct** handles characters. When **LC_CTYPE** is set to a valid value, **acct** can display and handle text and filenames containing valid characters for that locale. **acct** can display and handle Extended Unix Code (EUC) characters where any character can be 1, 2, or 3 bytes wide. **acct** can also handle EUC characters of 1, 2, or more column widths. In the "C" locale, only characters from ISO 8859-1 are valid.

LC_TIME determines how **acct** handles date and time formats. In the "C" locale, date and time handling follows the U.S. rules.

FILES

/etc/passwd used for login name to user ID conversions

/usr/lib/acct holds all accounting commands listed in sub-class 1M of this manual

/var/adm/pacct current process accounting file

/var/adm/wtmp login/logoff history file

SEE ALSO

acctcom(1), **acctcms(1M)**, **acctcon(1M)**, **acctmerg(1M)**, **acctprc(1M)**, **acctsh(1M)**, **fwtmp(1M)**, **runacct(1M)**, **acct(2)**, **acct(4)**, **utmp(4)**, **environ(5)**

System Administration Guide, Volume II

NOTES

The **acctdusg** command can process a maximum of

3000 distinct users

during a single invocation of the command. If at some point the actual number of users exceeds the maximum, the command will not succeed.

NAME	acctcms – command summary from process accounting records
SYNOPSIS	<code>/usr/lib/acct/acctcms [-a [-o] [-p]] [-c] [-j] [-n] [-s] [-t] filename ...</code>
DESCRIPTION	acctcms reads one or more <i>filenames</i> , normally in the form described in acct(4) . It adds all records for processes that executed identically-named commands, sorts them, and writes them to the standard output, normally using an internal summary format.
OPTIONS	<p>-a Print output in ASCII rather than in the internal summary format. The output includes command name, number of times executed, total kcore-minutes, total CPU minutes, total real minutes, mean size (in K), mean CPU minutes per invocation, "hog factor", characters transferred, and blocks read and written, as in acctcom(1). Output is normally sorted by total kcore-minutes.</p> <p>The following options may be used only with the -a option:</p> <p>-o Output a (non-prime) offshift-time-only command summary.</p> <p>-p Output a prime-time-only command summary.</p> <p>When -o and -p are used together, a combination prime-time and non-prime-time report is produced. All the output summaries will be total usage except number of times executed, CPU minutes, and real minutes, which will be split into prime and non-prime.</p> <p>-c Sort by total CPU time, rather than total kcore-minutes.</p> <p>-j Combine all commands invoked only once under "***other".</p> <p>-n Sort by number of command invocations.</p> <p>-s Any filenames encountered hereafter are already in internal summary format.</p> <p>-t Process all records as total accounting records. The default internal summary format splits each field into prime and non-prime time parts. This option combines the prime and non-prime time parts into a single field that is the total of both, and provides upward compatibility with old style acctcms internal summary format records.</p>
EXAMPLES	<p>A typical sequence for performing daily command accounting and for maintaining a running total is:</p> <pre>example% acctcms filename ... > today example% cp total previoustotal example% acctcms -s today previoustotal > example% acctcms -a -s today</pre>
SEE ALSO	acctcom(1) , acct(1M) , acctcon(1M) , acctmerg(1M) , acctprc(1M) , acctsh(1M) , fwtmp(1M) , runacct(1M) , acct(2) , acct(4) , utmp(4)

NOTES

Unpredictable output results if `-t` is used on new style internal summary format files, or if it is not used with old style internal summary format files.

NAME	acctcon, acctcon1, acctcon2 – connect-time accounting
SYNOPSIS	<pre> /usr/lib/acct/acctcon [-l <i>lineuse</i> [-o <i>reboot</i>] /usr/lib/acct/acctcon1 [-p] [-t] [-l <i>lineuse</i>] [-o <i>reboot</i>] /usr/lib/acct/acctcon2 </pre>
DESCRIPTION	<p>acctcon converts a sequence of login/logoff records to total accounting records (see the tacct format in acct(4)). The login/logoff records are read from standard input. The file /var/adm/wtmp is usually the source of the login/logoff records, however, because it may contain corrupted records or system date changes, it should first be fixed using wtmpfix. The fixed version of file /var/adm/wtmp can then be redirected to acctcon. The tacct records are written to standard output.</p> <p>acctcon is a combination of the programs acctcon1 and acctcon2. acctcon1 converts login/logoff records, taken from the fixed /var/adm/wtmp file, to ASCII output. acctcon2 reads the ASCII records produced by acctcon1 and converts them to tacct records. acctcon1 can be used with the -l and -o options, described below, as well as with the -p and -t options.</p>
OPTIONS	<p>-p Print input only, showing line name, login name, and time (in both numeric and date/time formats).</p> <p>-t acctcon1 maintains a list of lines on which users are logged in. When it reaches the end of its input, it emits a session record for each line that still appears to be active. It normally assumes that its input is a current file, so that it uses the current time as the ending time for each session still in progress. The -t flag causes it to use, instead, the last time found in its input, thus assuring reasonable and repeatable numbers for non-current files.</p> <p>-l <i>lineuse</i> <i>lineuse</i> is created to contain a summary of line usage showing line name, number of minutes used, percentage of total elapsed time used, number of sessions charged, number of logins, and number of logoffs. This file helps track line usage, identify bad lines, and find software and hardware oddities. Hangup, termination of login(1) and termination of the login shell each generate logoff records, so that the number of logoffs is often three to four times the number of sessions. See init(1M) and utmp(4).</p> <p>-o <i>reboot</i> <i>reboot</i> is filled with an overall record for the accounting period, giving starting time, ending time, number of reboots, and number of date changes.</p>
EXAMPLES	<p>The acctcon command is typically used as follows:</p> <pre> example% acctcon -l lineuse -o reboots < tmpwtmp > ctacct </pre> <p>The acctcon1 and acctcon2 commands are typically used as follows:</p> <pre> example% acctcon1 -l lineuse -o reboots < tmpwtmp sort +1n +2 > ctmp example% acctcon2 < ctmp > ctacct </pre>

FILES	<code>/var/adm/wtmp</code> login/logoff summary
SEE ALSO	acctcom(1) , login(1) , acct(1M) , acctcms(1M) , acctmerg(1M) , acctprc(1M) , acctsh(1M) , fwtmp(1M) , init(1M) , runacct(1M) , acct(2) , acct(4) , utmp(4) <i>System Administration Guide, Volume II</i>
NOTES	<p>The line usage report is confused by date changes. Use wtmpfix (see fwtmp(1M)), with the <code>/var/adm/wtmp</code> file as an argument, to correct this situation.</p> <p>The acctcon, acctcon1, and acctcon2 commands can process a maximum of</p> <ul style="list-style-type: none">6000 distinct sessions1000 distinct terminal lines2000 distinct login names <p>during a single invocation of any given command. If at some point the actual number of any one of these items exceeds the maximum, the command will not succeed.</p>

NAME	acctmrg – merge or add total accounting files
SYNOPSIS	<code>/usr/lib/acct/acctmrg [-a] [-i] [-p] [-t] [-u] [-v] [filename] ...</code>
DESCRIPTION	acctmrg reads its standard input and up to nine additional files, all in the tacct format (see acct(4)) or an ASCII version thereof. It merges these inputs by adding records whose keys (normally user ID and name) are identical, and expects the inputs to be sorted on those keys.
OPTIONS	-a Produce output in ASCII version of tacct . -i Input files are in ASCII version of tacct . -p Print input with no processing. -t Produce a single record that totals all input. -u Summarize by user ID, rather than by user ID and name. -v Produce output in verbose ASCII format, with more precise notation for floating-point numbers.
EXAMPLES	The following sequence is useful for making "repairs" to any file kept in this format: example% acctmrg -v <filename1 >filename2 Edit <i>filename2</i> as desired example% acctmrg -i <filename2 >filename1
SEE ALSO	acctcom(1) , acct(1M) , acctcms(1M) , acctcon(1M) , acctprc(1M) , acctsh(1M) , fwtmp(1M) , runacct(1M) , acct(2) , acct(4) , utmp(4) <i>System Administration Guide, Volume II</i>

NAME	acctprc, acctprc1, acctprc2 – process accounting
SYNOPSIS	<pre> /usr/lib/acct/acctprc /usr/lib/acct/acctprc1 [<i>ctmp</i>] /usr/lib/acct/acctprc2 </pre>
DESCRIPTION	<p>acctprc reads the standard input, in the form described by acct(4), and converts it to total accounting records (see the tacct record in acct(4)). acctprc divides CPU time into prime time and non-prime time and determines mean memory size (in memory segment units). acctprc then summarizes the tacct records, according to user IDs, and adds login names corresponding to the user IDs. The summarized records are then written to the standard output. acctprc1 reads input in the form described by acct(4), adds login names corresponding to user IDs, then writes for each process an ASCII line giving user ID, login name, prime CPU time (tics), non-prime CPU time (tics), and mean memory size (in memory segment units). If <i>ctmp</i> is given, it is expected to contain a list of login sessions sorted by user ID and login name. If this file is not supplied, it obtains login names from the password file, just as acctprc does. The information in <i>ctmp</i> helps it distinguish between different login names sharing the same user ID.</p> <p>From the standard input, acctprc2 reads records in the form written by acctprc1, summarizes them according to user ID and name, then writes the sorted summaries to the standard output as total accounting records.</p>
EXAMPLES	<p>The acctprc command is typically used as shown below:</p> <pre>example% acctprc < /var/adm/pacct > ptacct</pre> <p>The acctprc1 and acctprc2s commands are typically used as shown below:</p> <pre>example% acctprc1 <i>ctmp</i> </var/adm/pacct example% acctprc2 > ptacct</pre>
FILES	<i>/etc/passwd</i> system password file
SEE ALSO	acctcom(1) , acct(1M) , acctcms(1M) , acctcon(1M) , acctmerg(1M) , acctsh(1M) , cron(1M) , fwtmp(1M) , runacct(1M) , acct(2) , acct(4) , utmp(4)
NOTES	<p>Although it is possible for acctprc1 to distinguish among login names that share user IDs for commands run normally, it is difficult to do this for those commands run from cron(1M), for example. A more precise conversion can be done using the acctwtmp program in acct(1M). acctprc does not distinguish between users with identical user IDs.</p> <p>A memory segment of the mean memory size is a unit of measure for the number of bytes in a logical memory segment on a particular processor.</p>

The **acctprc**, **acctprc1**, and **acctprc2** commands can process a maximum of

- 6000 distinct sessions
- 1000 distinct terminal lines
- 2000 distinct login names

during a single invocation of any given command. If at some point the actual number of any one of these items exceeds the maximum, the command will not succeed.

NAME	acctsh, chargefee, ckpacct, dodisk, lastlogin, monacct, nulladm, prctmp, prdaily, prtacct, runacct, shutacct, startup, turnacct – shell procedures for accounting
SYNOPSIS	<pre> /usr/lib/acct/chargefee login-name number /usr/lib/acct/ckpacct [blocks] /usr/lib/acct/dodisk [-o] [filename ...] /usr/lib/acct/lastlogin /usr/lib/acct/monacct number /usr/lib/acct/nulladm filename ... /usr/lib/acct/prctmp filename /usr/lib/acct/prdaily [-c] [-l] [mmdd] /usr/lib/acct/prtacct filename ["heading"] /usr/lib/acct/runacct [mmdd] [mmdd state] /usr/lib/acct/shutacct ["reason"] /usr/lib/acct/startup /usr/lib/acct/turnacct on off switch </pre>
DESCRIPTION	
chargefee Command	chargefee can be invoked to charge a <i>number</i> of units to <i>login-name</i> . A record is written to <i>/var/adm/fee</i> , to be merged with other accounting records by runacct .
ckpacct Command	ckpacct should be initiated using cron (1M) to periodically check the size of <i>/var/adm/pacct</i> . If the size exceeds <i>blocks</i> , 500 by default, turnacct will be invoked with argument switch . If the number of free disk blocks in the <i>/var</i> file system falls below 500 , ckpacct will automatically turn off the collection of process accounting records via the off argument to turnacct . When at least 500 blocks are restored, the accounting will be activated again on the next invocation of ckpacct . This feature is sensitive to the frequency at which ckpacct is executed, usually by cron .
dodisk Command	dodisk should be invoked by cron to perform the disk accounting functions.
lastlogin Command	lastlogin is invoked by runacct to update <i>/var/adm/acct/sum/loginlog</i> , which shows the last date on which each person logged in.
monacct Command	monacct should be invoked once each month or each accounting period. <i>number</i> indicates which month or period it is. If <i>number</i> is not given, it defaults to the current month (01–12). This default is useful if monacct is to be executed using cron (1M) on the first day of each month. monacct creates summary files in <i>/var/adm/acct/fiscal</i> and restarts the summary files in <i>/var/adm/acct/sum</i> .

nulladm Command	nulladm creates <i>filename</i> with mode 664 and ensures that owner and group are adm . It is called by various accounting shell procedures.
prctmp Command	prctmp can be used to print the session record file (normally <i>/var/adm/acct/nite/ctmp</i> created by acctcon1 (see acctcon(1M)).
prdaily Command	prdaily is invoked by runacct to format a report of the previous day's accounting data. The report resides in <i>/var/adm/acct/sum/rprt/mmdd</i> where <i>mmdd</i> is the month and day of the report. The current daily accounting reports may be printed by typing prdaily . Previous days' accounting reports can be printed by using the <i>mmdd</i> option and specifying the exact report date desired.
prtacct Command	prtacct can be used to format and print any total accounting (taacct)file.
runacct Command	runacct performs the accumulation of connect, process, fee, and disk accounting on a daily basis. It also creates summaries of command usage. For more information, see runacct(1M) .
shutacct Command	shutacct is invoked during a system shutdown to turn process accounting off and append a <i>reason</i> record to <i>/var/adm/wtmp</i> .
startup Command	startup can be invoked when the system is brought to a multi-user state to turn process accounting on.
turnacct Command	turnacct is an interface to accton (see acct(1M)) to turn process accounting on or off . The switch argument moves the current <i>/var/adm/pacct</i> to the next free name in <i>/var/adm/pacctincr</i> (where <i>incr</i> is a number starting with 1 and incrementing by one for each additional pacct file), then turns accounting back on again. This procedure is called by ckpacct and thus can be taken care of by the cron and used to keep pacct to a reasonable size. shutacct uses turnacct to stop process accounting. startup uses turnacct to start process accounting.
OPTIONS	<ul style="list-style-type: none"> -c This option prints a report of exceptional resource usage by command, and may be used on current day's accounting data only. -l This option prints a report of exceptional usage by login id for the specified date. Previous daily reports are cleaned up and therefore inaccessible after each invocation of monacct. -o Use acctdusg (see acct(1M)) to do a slower version of disk accounting by login directory. <i>filenames</i> specifies the one or more filesystem names where disk accounting will be done. If <i>filenames</i> are used, disk accounting will be done on these filesystems only. If the -o option is used, <i>filenames</i> should be mount points of mounted filesystems. If the -o option is omitted, <i>filenames</i> should be the special file names of mountable filesystems.

FILES	<p>/usr/lib/acct holds all accounting commands listed in section 1M of this manual</p> <p>/usr/lib/acct/ptecms.awk contains the limits for exceptional usage by command name</p> <p>/usr/lib/acct/ptelus.awk contains the limits for exceptional usage by login ID</p> <p>/var/adm/acct/fiscal fiscal reports directory</p> <p>/var/adm/acct/nite working directory</p> <p>/var/adm/acct/sum summary directory contains information for monacct</p> <p>/var/adm/acct/sum/loginlog file updated by last login</p> <p>/var/adm/fee accumulator for fees</p> <p>/var/adm/pacct current file for per-process accounting</p> <p>/var/adm/pacctincr used if pacct gets large and during execution of daily accounting procedure</p> <p>/var/adm/wtmp login/logoff summary</p>
SEE ALSO	acctcom(1) , acct(1M) , acctcms(1M) , acctcon(1M) , acctmerg(1M) , acctprc(1M) , cron(1M) , fwtmp(1M) , runacct(1M) , acct(2) , acct(4) , utmp(4)
NOTES	<p>The dodisk command can process a maximum of 3000 distinct users during a single invocation of the command. If at some point the actual number of users exceeds the maximum, the command will not succeed.</p>

NAME	adbgen – generate adb script
SYNOPSIS	<code>/usr/lib/adb/adbgen filename.adb ...</code>
AVAILABILITY	SUNWesu
DESCRIPTION	<p>adbgen makes it possible to write adb(1) scripts that do not contain hard-coded dependencies on structure member offsets. The input to adbgen is a file named <i>filename.adb</i> which contains header information, then a null line, then the name of a structure, and finally an adb script. adbgen only deals with one structure per file; all member names are assumed to be in this structure. The output of adbgen is an adb script in <i>filename</i>. adbgen operates by generating a C program which determines structure member offsets and sizes, which in turn generate the adb script.</p> <p>The header lines, up to the null line, are copied verbatim into the generated C program. Typically these are #include statements, which include the headers containing the relevant structure declarations.</p> <p>The adb script part may contain any valid adb commands (see adb(1)), and may also contain adbgen requests, each enclosed in {}s. Request types are:</p> <ul style="list-style-type: none"> • Print a structure member. The request form is {<i>member, format</i>}. <i>member</i> is a member name of the <i>structure</i> given earlier, and <i>format</i> is any valid adb format request. For example, to print the p_pid field of the <i>proc</i> structure as a decimal number, you would write {p_pid,d}. • Reference a structure member. The request form is {*<i>member, base</i>}. <i>member</i> is the member name whose value is desired, and <i>base</i> is an adb register name which contains the base address of the structure. For example, to get the p_pid field of the <i>proc</i> structure, you would get the <i>proc</i> structure address in an adb register, for example <f, and write {*p_pid,<f}. • Tell adbgen that the offset is ok. The request form is {OFFSETOK}. This is useful after invoking another adb script which moves the adb dot. • Get the size of the <i>structure</i>. The request form is {SIZEOF}. adbgen replaces this request with the size of the structure. This is useful in incrementing a pointer to step through an array of structures. • Calculate an arbitrary C expression. The request form is {EXPR, expression}. adbgen replaces this request with the value of the expression. This is useful when more than one structure is involved in the script. • Get the offset to the end of the structure. The request form is {END}. This is useful at the end of the structure to get adb to align the <i>dot</i> for printing the next structure member. <p>adbgen keeps track of the movement of the adb dot and generates adb code to move forward or backward as necessary before printing any structure member in a script. adbgen's model of the behavior of adb's <i>dot</i> is simple: it is assumed that the first line of the script is of the form <i>struct_address/adb text</i> and that subsequent lines are of the form</p>

+/*adb text*. The **adb dot** then moves in a sane fashion. **adbgen** does not check the script to ensure that these limitations are met. **adbgen** also checks the size of the structure member against the size of the **adb** format code and warns you if they are not equal.

EXAMPLES

If there were an include file **x.h** which contained

```
struct x {
    char    *x_cp;
    char    x_c;
    int     x_i;
};
```

then an **adbgen** file (call it **script.adb**) to print the file **x.h** would be:

```
#include "x.h"

x
./"x_cp"16t"x_c"8t"x_i"n{x_cp,X}{x_c,C}{x_i,D}
```

After running **adbgen** the output file **script** would contain:

```
16t"x_c"8t"x_i"nXC3+D"" ./"x_cp"16t"x_c"8t"x_i"nXC3+D
```

To invoke the script you would type:

```
example% adb program
x$<script
```

FILES

*/usr/platform/platform-name/lib/adb/** platform-specific **adb** scripts for debugging the kernel
*/usr/lib/adb/** **adb** scripts for debugging the kernel

SEE ALSO

adb(1), **kadb(1M)**

DIAGNOSTICS

Warnings are given about structure member sizes not equal to **adb** format items and about badly formatted requests. The C compiler complains if you reference a structure member that does not exist. It also complains about an ampersand before array names; these complaints may be ignored.

NOTES

platform-name can be found using the **-i** option of **uname(1)**.

BUGS

adb syntax is ugly; there should be a higher level interface for generating scripts. Structure members which are bit fields cannot be handled because C will not give the address of a bit field. The address is needed to determine the offset.

NAME	add_drv – add a new device driver to the system
SYNOPSIS	add_drv [-b <i>basedir</i>] [-c <i>class_name</i>] [-i ' <i>identify_name...</i> '] [-m ' <i>permission',...</i> '] [-n] <i>device_driver</i>
AVAILABILITY	SUNWcsu
DESCRIPTION	The add_drv command is used to inform the system about newly installed device drivers.
OPTIONS	<p>-b <i>basedir</i> Sets the path to the root directory of the diskless client. Used on the server to execute add_drv for a diskless client. The client machine must be rebooted to install the driver.</p> <p>-c <i>class_name</i> The driver being added to the system exports the class <i>class_name</i>.</p> <p>-i '<i>identify_name</i>' A white-space separated list of aliases for the driver <i>device_driver</i>.</p> <p>-m '<i>permission</i>' Specify the file system permissions for device nodes created by the system on behalf of <i>device_driver</i>.</p> <p>-n Do not try to load and attach the driver, just modify the system configuration files for the <i>device_driver</i>.</p>
EXAMPLES	<p>The following example adds the SUNW,example driver to the system, with an alias name of SUNW,alias. It assumes the driver has already been copied to /usr/kernel/drv.</p> <pre>example# add_drv -m '* 0666 bin bin', 'a 0644 root sys' \ -i 'SUNW,alias' SUNW,example</pre> <p>Every minor node created by the system for the SUNW,example driver will have the permission 0666, and be owned by user bin in the group bin, except for the minor device a, which will be owned by root, group sys, and have a permission of 0644.</p> <p>The following example adds the driver to the client /export/root/sun1. The driver is installed and loaded when the client machine, sun1, is rebooted.</p> <pre>example# add_drv -m '* 0666 bin bin', 'a 0644 root sys' \ -i 'SUNW,alias' -b /export/root/sun1 \ SUNW,example</pre> <p>This second example produces the same result as the first, except the changes are on the diskless client, sun1, and the client must be rebooted for the driver to be installed.</p>
EXIT CODES	add_drv returns 0 on success and 1 on failure.
FILES	<p>/kernel/drv boot device drivers, platform-dependent drivers.</p> <p>/usr/kernel/drv other drivers that could potentially be shared between platforms.</p> <p>/etc/driver_aliases driver aliases file.</p> <p>/etc/driver_classes driver classes file.</p> <p>/etc/minor_perm minor node permissions.</p>

`/etc/name_to_major` major number binding.

SEE ALSO [devlinks\(1M\)](#), [disks\(1M\)](#), [drvconfig\(1M\)](#), [kernel\(1M\)](#), [modinfo\(1M\)](#), [ports\(1M\)](#), [rem_drv\(1M\)](#), [tapes\(1M\)](#), [driver.conf\(4\)](#), [system\(4\)](#), [ddi_create_minor_node\(9F\)](#)

Writing Device Drivers

NOTES Aliases may require quoting (with double-quotes) if they contain numbers.

BUGS `add_drv` will accept a full pathname for *device_driver*. However, the kernel does not use the full pathname; it only uses the final component and searches the internal driver search path for the driver. This can lead to the kernel loading a different driver than expected.

For this reason, it is **not** recommended that you use `add_drv` with a full pathname. See [kernel\(1M\)](#) for more information on the driver search path.

NAME	addbadsec – map out defective disk blocks
SYNOPSIS	addbadsec [-p] [-a <i>blkno</i> [<i>blkno</i> ...]] [-f <i>filename</i>] <i>raw_device</i>
AVAILABILITY	x86 SUNWcsu
DESCRIPTION	<p>addbadsec is used by the system administrator to map out bad disk blocks. Normally, these blocks are identified during surface analysis, but occasionally the disk subsystem reports unrecoverable data errors indicating a bad block. A block number reported in this way can be fed directly into addbadsec, and the block will be remapped. addbadsec will first attempt hardware remapping. This is supported on SCSI drives and takes place at the disk hardware level. If the target is an IDE drive, then software remapping is used. In order for software remapping to succeed, the partition must contain an alternate slice and there must be room in this slice to perform the mapping.</p> <p>-p Causes addbadsec to print the current software map. The output shows the defective block and the assigned alternate. This option cannot be used to print the hardware map.</p> <p>-a Adds the specified blocks to the hardware or software map. If more than one block number is specified, the entire list should be quoted and block numbers should be separated by white space.</p> <p>-f Adds the specified blocks to the hardware or software map. The bad blocks are listed, one per line, in the specified file.</p> <p>It should be understood that bad blocks lead to data loss. Remapping a defective block does not repair a damaged file. If a bad block occurs to a disk-resident file system structure such as a superblock, the entire slice might have to be recovered from a backup.</p>
FILES	The raw device should be /dev/rdsk/c?t?d?p0 .
SEE ALSO	fdisk(1M) , fmthard(1M) , diskscan(1M)

NAME	admintool – system administration with a graphical user interface												
SYNOPSIS	/usr/bin/admintool												
AVAILABILITY	SUNWadmap												
DESCRIPTION	<p>admintool is a graphical user interface that enables you to accomplish several system administration tasks on a local system. Membership in the sysadmin group (gid 14) is used to restrict access to administrative tasks. Members of the sysadmin group can use admintool to create, delete, and modify local system files. Non-members have read-only permissions (where applicable).</p> <p>Help is available by using the Help button.</p> <p>Note: admintool is not the tool for a distributed environment. It is used for local administration.</p>												
USAGE	<p>admintool allows you to do the following tasks:</p> <table border="0"> <tr> <td style="padding-right: 20px;">Manage users</td> <td>Use admintool to add, delete, or modify user accounts. admintool makes the appropriate changes to the system's /etc/passwd file (see passwd(4)).</td> </tr> <tr> <td>Manage groups</td> <td>Use admintool to add, delete, or modify groups. admintool makes the appropriate changes to the system's /etc/group file (see group(4)).</td> </tr> <tr> <td>Manage hosts</td> <td>Use admintool to add, delete, or modify hosts. admintool makes the appropriate changes to the system's /etc/hosts file (see hosts(4)).</td> </tr> <tr> <td>Manage printers</td> <td>Use admintool to add or delete access to a printer, or to modify a system's printer access. admintool makes the appropriate changes to the system's /etc/lp directory.</td> </tr> <tr> <td>Manage serial port services</td> <td>Use admintool to enable or disable serial port services. admintool sets up the software services necessary to use a modem or terminal attached to a system's serial port.</td> </tr> <tr> <td>Manage software</td> <td>Use admintool to add or remove software. admintool adds software from a product CD or on a hard disk to an installed system, or removes software from an installed system.</td> </tr> </table>	Manage users	Use admintool to add, delete, or modify user accounts. admintool makes the appropriate changes to the system's /etc/passwd file (see passwd(4)).	Manage groups	Use admintool to add, delete, or modify groups. admintool makes the appropriate changes to the system's /etc/group file (see group(4)).	Manage hosts	Use admintool to add, delete, or modify hosts. admintool makes the appropriate changes to the system's /etc/hosts file (see hosts(4)).	Manage printers	Use admintool to add or delete access to a printer, or to modify a system's printer access. admintool makes the appropriate changes to the system's /etc/lp directory.	Manage serial port services	Use admintool to enable or disable serial port services. admintool sets up the software services necessary to use a modem or terminal attached to a system's serial port.	Manage software	Use admintool to add or remove software. admintool adds software from a product CD or on a hard disk to an installed system, or removes software from an installed system.
Manage users	Use admintool to add, delete, or modify user accounts. admintool makes the appropriate changes to the system's /etc/passwd file (see passwd(4)).												
Manage groups	Use admintool to add, delete, or modify groups. admintool makes the appropriate changes to the system's /etc/group file (see group(4)).												
Manage hosts	Use admintool to add, delete, or modify hosts. admintool makes the appropriate changes to the system's /etc/hosts file (see hosts(4)).												
Manage printers	Use admintool to add or delete access to a printer, or to modify a system's printer access. admintool makes the appropriate changes to the system's /etc/lp directory.												
Manage serial port services	Use admintool to enable or disable serial port services. admintool sets up the software services necessary to use a modem or terminal attached to a system's serial port.												
Manage software	Use admintool to add or remove software. admintool adds software from a product CD or on a hard disk to an installed system, or removes software from an installed system.												
EXIT STATUS	admintool terminates with exit status 0 .												
SEE ALSO	<p>group(4), hosts(4), passwd(4)</p> <p><i>Solaris Advanced User's Guide</i></p>												

WARNINGS

If you use **admintool** to add a host, your local system and your site uses a network name service such as NIS or NIS+, **admintool** host operations may not have the desired effect. This is because information in the network name service will take precedence over the information in the local `/etc/hosts` file, which is where **admintool** updates information.

NOTES

admintool modifies files on the local system, i.e., the system on which you are running **admintool**. **admintool** does not modify or update global networked databases such as NIS or NIS+.

NAME	aliasadm – manipulate the NIS+ aliases map
SYNOPSIS	aliasadm –a <i>alias expansion</i> [<i>options comments</i>] [<i>optional flags ...</i>] aliasadm –c <i>alias expansion</i> [<i>options comments</i>] [<i>optional flags ...</i>] aliasadm –d <i>alias</i> [<i>optional flags ...</i>] aliasadm –e <i>alias</i> [<i>optional flags ...</i>] aliasadm –l <i>alias</i> [<i>optional flags ...</i>] aliasadm –m <i>alias</i> [<i>optional flags ...</i>] aliasadm [–I] [–D <i>domainname</i>] [–f <i>filename</i>] [–M <i>mapname</i>]
DESCRIPTION	<p>aliasadm makes changes to the alias map.</p> <p>The alias map is an NIS+ table object with four columns:</p> <p><i>alias</i> The name of the alias as a null terminated string.</p> <p><i>expansion</i> The value of the alias as it would appear in a sendmail /etc/aliases file.</p> <p><i>options</i> A list of options applicable to this alias. The only option currently supported is CANON. With this option, if the user has requested an inverse alias lookup, and there is more than one alias with this expansion, this alias is given preference.</p> <p><i>comments</i> An arbitrary string containing comments about this alias. The sendmail(1M) command reads this map in addition to the NIS aliases map and the local /etc/aliases database.</p>
OPTIONS	–a Add an alias. –c Change an alias. –d Delete an alias. –e Edit the alias map. –I Initialize the NIS+ aliases database. –l List the alias map. –m Print or match an alias. –D <i>domainname</i> Edit the map in domain <i>domainname</i> instead of the current domain. –f <i>filename</i> When editing or listing the database, use <i>filename</i> instead of invoking the editor. –M <i>mapname</i> Edit <i>mapname</i> instead of <i>mail_aliases</i> .
FILES	/etc/aliases mail aliases for the local host in ASCII format
SEE ALSO	sendmail (1M)

NAME	allocate – device allocation
SYNOPSIS	allocate [-s] [-U <i>uname</i>] <i>device</i> allocate [-s] [-U <i>uname</i>] -g <i>dev-type</i> allocate [-s] [-U <i>uname</i>] -F <i>device</i>
AVAILABILITY	The functionality described in this man page is available only if the Basic Security Module (BSM) has been enabled. See bsmconv (1M) for more information.
DESCRIPTION	<p>allocate manages the ownership of devices through its allocation mechanism. It ensures that each device is used by only one qualified user at a time.</p> <p>The <i>device</i> argument specifies the device to be manipulated. To preserve the integrity of the device's owner, the allocate operation is executed on all the device special files associated with that device.</p> <p>The argument <i>dev-type</i>, is the device type to be operated on. The argument <i>dev-type</i>, can only be used with the -g option.</p> <p>The default allocate operation, allocates the device special files associated with <i>device</i> to the uid of the current process.</p> <p>If the -F option is specified, the device cleaning program is executed when allocation is performed. This cleaning program is found in /etc/security/lib. The name of this program is found in the device_allocate(4) entry for the device in the <i>dev-exec</i> field.</p>
OPTIONS	<p>-g <i>dev-type</i> Allocate a non-allocated device with a device-type matching <i>dev-type</i>.</p> <p>-s Silent. Suppresses any diagnostic output.</p> <p>-F <i>device</i> Reallocate the device allocated to another user. This option is often used with -U to reallocate a specific device to a specific user. Only the super user is permitted to use this option.</p> <p>-U <i>uname</i> Use the user ID <i>uname</i> instead of the user ID of the current process when performing the allocate operation. Only the super user is permitted to use this option.</p>
DIAGNOSTICS	allocate returns a nonzero exit status in the event of an error.
FILES	/etc/security/device_allocate /etc/security/device_maps /etc/security/dev/* /etc/security/lib/*
SEE ALSO	bsmconv (1M), device_allocate (4), device_maps (4)

NAME	arp – address resolution display and control
SYNOPSIS	arp <i>hostname</i> arp -a arp -d <i>hostname</i> arp -f <i>filename</i> arp -s <i>hostname ether_address</i> [temp] [pub] [trail]
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>The arp program displays and modifies the Internet-to-Ethernet address translation tables used by the address resolution protocol (see arp(7P)).</p> <p>With no flags, the program displays the current ARP entry for <i>hostname</i>. The host may be specified by name or by number, using Internet dot notation.</p>
OPTIONS	<p>-a Display all of the current ARP entries. The definition for the flags in the table are:</p> <p style="padding-left: 2em;">P Publish; includes IP address for the machine and the addresses that have explicitly been added by the -s option. ARP will respond to ARP requests for this address.</p> <p style="padding-left: 2em;">S Static; not learned for the ARP protocol.</p> <p style="padding-left: 2em;">U Unresolved; waiting for ARP response.</p> <p style="padding-left: 2em;">M Mapping; only used for the multicast entry for 224.0.0.0</p> <p>-d Delete an entry for the host called <i>hostname</i>. This option may only be used by the super-user.</p> <p>-f Read the file named <i>filename</i> and set multiple entries in the ARP tables. Entries in the file should be of the form</p> <p style="padding-left: 4em;"><i>hostname ether_address</i> [temp] [pub] [trail]</p> <p>(see option -s for argument definitions).</p> <p>-s Create an ARP entry for the host called <i>hostname</i> with the Ethernet address <i>ether_address</i>. The Ethernet address is given as six hexadecimal bytes separated by colons. The entry will be permanent unless the word temp is given in the command. If the word pub is given, the entry will be published. For instance, this system will respond to ARP requests for <i>hostname</i> even though the hostname is not its own. The word trail indicates that trailer encapsulations may be sent to this host. arp -s can be used for a limited form of proxy ARP when a host on one of the directly attached networks is not physically present on the subnet. Another machine can then be configured to respond to ARP requests using arp -s. This is useful in certain SLIP or PPP configurations.</p>

SEE ALSO | **ifconfig(1M), arp(7P)**

NAME	aset – monitors or restricts accesses to system files and directories
SYNOPSIS	aset [-p] [-d <i>aset_dir</i>] [-l <i>sec_level</i>] [-n <i>user@host</i>] [-u <i>userlist_file</i>]
DESCRIPTION	<p>The Automated Security Enhancement Tool (ASET) is a set of administrative utilities that can improve system security by allowing the system administrators to check the settings of system files, including both the attributes (permissions, ownership, etc.) and the contents of the system files. It warns the users of potential security problems and, where appropriate, sets the system files automatically according to the security level specified.</p> <p>The security level for aset can be specified by setting the -l command line option or the ASETSECLEVEL environment variable to be one of 3 values: low, med, or high. All the functionality operates based on the value of the security level.</p> <p>At the low level, aset does not alter any system behavior. It merely performs a number of checks and reports any potential security weaknesses.</p> <p>At the med level, aset modifies some of the settings of system files and parameters, thus restricting system access, to reduce the risks from security attacks. Again it will report the security weaknesses and the modifications performed to restrict access. This does not affect the operations of system services. All the system applications and commands will maintain all of their original functionality.</p> <p>At the high level, further restrictions are made to system access, rendering a very defensive system. Security practices which are not normally required are included. Many system files and parameters settings are modified to minimum access permissions. At this level, security is the foremost concern, higher than any other considerations that affect system behavior. The vast majority of system applications and commands will maintain their functionality, although there may be a few that exhibit behaviors that are not familiar in normal system environment.</p> <p>More exact definitions of these levels (what exactly aset will do at each level) can be found in the administrator manual. The asetenv(4) file and the <i>master files</i> (see asetmasters(4)) determine to a large extent what aset performs at each level, and can be used by the experienced administrators to redefine the definitions of the levels to suit their particular needs. These files are provided by default to fit most security conscious environments and in most cases provide adequate security safeguards without modification. They are, however, designed in a way that can be easily edited by experienced administrators with specific needs.</p> <p>aset can be periodically activated at the specified security level with default definitions using the -p option. aset will be automatically activated at a frequency specified by the administrator starting from a designated future time (see asetenv(4)). Without the -p option, aset will operate only once immediately.</p>

OPTIONS	<p>-p Schedule aset to be executed periodically. This adds an entry for aset in the /etc/crontab file. The PERIODIC_SCHEDULE environment variable in the /usr/aset/asetenv file is used to define the time for execution. See crontab(1) and asetenv(4). If a crontab(1) entry for aset already exists, a warning is produced in the execution log.</p> <p>-d <i>aset_dir</i> Specify a working directory other than /usr/aset for ASET. /usr/aset is the default working directory. It is where ASET is installed, and is the <i>root</i> directory of all ASET utilities and data files. If another directory is to be used as the ASET working directory you can either define it with the -d option, or by setting the ASETDIR environment variable before invoking aset. The command line option, if specified, overwrites the environment variable.</p> <p>-l <i>sec_level</i> Specify a security level (low, med, or high) for aset to operate at. The default level is low. Each security level is explained in detail above. The level can also be specified by setting the ASETSECLEVEL environment variable before invoking aset. The command line option, if specified, overwrites the environment variable.</p> <p>-n <i>user@host</i> Notify <i>user</i> at machine <i>host</i>. Send the output of aset to <i>user</i> through e-mail. If this option is not specified, the output is sent to the standard output. Note that this is not the reports of ASET, but rather an execution log including error messages if there are any. This output is typically fairly brief. The actual reports of ASET are found in the /usr/aset/reports/latest directory. See the -d option.</p> <p>-u <i>userlist_file</i> Specify a file containing a list of users. aset will perform environment checks (for example, UMASK and PATH variables) on these users. By default, aset only checks for root. <i>userlist_file</i> is an ASCII text file. Each entry in the file is a line that contains only one user name (login name).</p>
USAGE	<p>The following paragraphs discuss the features provided by ASET. Hereafter, each feature is referred to as a <i>task</i>. The first task, tune, is intended to be executed only once per installation of ASET. The other tasks are intended to be executed periodically at the specified frequency.</p>
tune Task	<p>This task is used to tighten system file permissions. In standard releases, system files or directories have permissions defined to maximize open information sharing. In a more security conscious environment, the administrator may want to redefine these permission settings to more restrictive values. aset allows resetting of these permissions, based on the specified security level. Generally, at the low level the permissions</p>

are set to what they should be as released. At the medium level the permissions are tightened to ensure reasonable security that is adequate for most environments. At the high level they are further tightened to very restrictive access. The system files affected and the respective restrictions at different levels are configurable, using the **tune.low**, **tune.med**, and **tune.high** files. See **asetmasters(4)**.

cklist Task

System directories that contain relatively static files (that is, their contents and attributes do not change frequently) are examined and compared with a master description file. The **/usr/aset/masters/cklist.level** files are automatically generated the first time the **cklist** task is executed. See **asetenv(4)**. Any discrepancy found is reported. The directories and files are compared based on the following:

- owner and group
- permission bits
- size and checksum (if file)
- number of links
- last modification time

The lists of directories to check are defined in **asetenv(4)**, based on the specified security level, and are configurable using the **CKLISTPATH_LOW**, **CKLISTPATH_MED**, and **CKLISTPATH_HIGH** environment variables. Typically, the lower level lists are subsets of the higher level lists.

usrgrp Task

aset checks the consistency and integrity of user accounts and groups as defined in the **passwd** and **group** databases, respectively. Any potential problems are reported. Potential problems for the **passwd** file include:

- **passwd** file entries are not in the correct format.
- User accounts without a password.
- Duplicate user names.
- Duplicate user IDs. Duplicate user IDs are reported unless allowed by the **uid_alias** file. See **asetmasters(4)**.
- Invalid login directories.
- If C2 is enabled, check C2 hidden **passwd** format.

Potential problems for the group file include:

- Group file entries not in the right format.
- Duplicate group names.
- Duplicate group IDs.
- Null group passwords.

aset checks the local **passwd** file. If the **YPCHECK** environment variable is set to **true**, **aset** also checks the NIS **passwd** files. See **asetenv(4)**. Problems in the NIS **passwd** file are only reported and not corrected automatically. The checking is done for all three security levels except where noted.

sysconf Task

aset checks various system configuration tables, most of which are in the `/etc` directory. **aset** checks and makes appropriate corrections for each system table at all three levels except where noted. The following discussion assumes familiarity with the various system tables. See the manual pages for these tables for further details.

The operations for each system table are:

/etc/hosts.equiv	The default file contains a single "+" line, thus making every known host a trusted host, which is not advised for system security. aset performs the following operations:
	<p>Low Warns the administrators about the "+" line.</p>
	<p>Medium</p>
	<p>High Warns about and deletes that entry.</p>
/etc/inetd.conf	The following entries for system daemons are checked for possible weaknesses.
	<p>ftpd(1) does not do any authentication. aset ensures that in.ftpd(1M) is started in the right directory on the server and is not running on clients. At the low level, it gives warnings if the mentioned condition is not true. At the medium and high levels it gives warnings, and changes (if necessary) the in.ftpd entry to include the <code>-s /tftpboot</code> option after ensuring the directory <code>/tftpboot</code> exists.</p>
	<p>ps(1) and netstat(1M) provide valuable information to potential system crackers. These are disabled when aset is executed at a high security level.</p>
	<p>rexid is also known to have poor authentication mechanism. aset disables rexid for medium and high security levels by commenting out this entry. If rexid is activated with the <code>-s</code> (secure RPC) option, it is not disabled.</p>
/etc/aliases	The decode alias of UUCP is a potential security weakness. aset disables the alias for medium and high security levels by commenting out this entry.
/etc/default/login	The <code>CONSOLE=</code> line is checked to allow root login only at a specific terminal depending on the security level:
	<p>Low No action taken.</p>
	<p>Medium</p>
	<p>High Adds the following line to the file: <code>CONSOLE=/dev/console</code></p>
/etc/vfstab	aset checks for world-readable or writeable device files for mounted file systems.
/etc/dfs/dfstab	aset checks for file systems that are exported without any restrictions.

	/etc/ftusers	At high security level, aset ensures root is in /etc/ftusers (create if necessary), thus disallowing ftp(1) to be used as root.
	/var/adm/utmp /var/adm/utmpx	aset makes these files not world-writeable for the high level (some applications may not run properly with this setting.)
	/.rhosts	The usage of a .rhosts file for the entire system is not advised. aset gives warnings for the low level and moves it to /.rhosts.bak for levels medium and high.
env Task		aset checks critical environment variables for root and users specified with the -u userlist_file option by parsing the /.profile , /.login , and /.cshrc files. This task checks the PATH variable to ensure that it does not contain '.' as a directory, which makes an easy target for <i>trojan horse</i> attacks. It also checks that the directories in the PATH variable are not world-writeable. Furthermore, it checks the UMASK variable to ensure files are not created as readable or writeable by world. Any problems found by these checks are reported.
eeprom Task		Newer versions of the EEPROM allow specification of a secure parameter. See eeprom(1M) . aset recommends that the administrator sets the parameter to command for the medium level and to full for the high level. It gives warnings if it detects the parameter is not set adequately.
firewall Task		At the high security level, aset takes proper measures such that the system can be safely used as a firewall in a network. This mainly involves disabling IP packets forwarding and making routing information invisible. Firewalling provides protection against external access to the network.
ENVIRONMENT	ASETDIR	Specify ASET's working directory. Defaults to /usr/aset .
	ASETSECLEVEL	Specify ASET's security level. Defaults to low .
	TASKS	Specify the tasks to be executed by aset . Defaults to all tasks .
FILES	/usr/aset/reports	directory of ASET reports
SEE ALSO	crontab(1) , ftp(1) , ps(1) , tftp(1) , eeprom(1M) , in.tftpd(1M) , netstat(1M) , asetenv(4) , aset-masters(4) <i>System Administration Guide, Volume I</i>	

NAME	aset.restore – restores system files to their content before ASET is installed
SYNOPSIS	aset.restore [-d <i>aset_dir</i>]
DESCRIPTION	<p>aset.restore restores system files that are affected by the Automated Security Enhancement Tool (ASET) to their pre-ASET content. When ASET is executed for the first time, it saves and archives the original system files in the /usr/aset/archives directory. The aset.restore utility reinstates these files. It also deschedules ASET, if it is currently scheduled for periodic execution. See asetenv(4).</p> <p>Note: If you have made changes to system files after running ASET, these changes are lost when you run aset.restore. If you want to be absolutely sure that you keep the existing system state, it is recommended that you back-up your system before using aset.restore.</p> <p>You should use aset.restore, under the following circumstances:</p> <ul style="list-style-type: none">You want to remove ASET permanently and restore the original system (if you want to deactivate ASET, you can remove it from scheduling).You are unfamiliar with ASET and want to experiment with it. You can use aset.restore to restore the original system state.When some major system functionality is not working properly and you suspect that ASET is causing the problem; you may want to restore the system to see if the problem persists without ASET. <p>aset.restore requires root privileges to execute.</p>
OPTIONS	-d <i>aset_dir</i> Specify the working directory for ASET. By default, this directory is /usr/aset . With this option the archives directory will be located under <i>aset_dir</i> .
FILES	/usr/aset/archives archive of system files prior to executing aset
SEE ALSO	aset(1M) , asetenv(4) <i>System Administration Guide, Volume I</i>

NAME	aspppd, aspppls – Asynchronous PPP Link Manager
SYNOPSIS	<code>/usr/sbin/aspppd [-d <i>debug-level</i>]</code> <code>/usr/sbin/aspppls</code>
AVAILABILITY	SUNWapppu
DESCRIPTION	<p>aspppd is the link manager for the asynchronous data link protocol specified in RFC1331, <i>The Point-to-Point Protocol (PPP) for the Transmission of Multi-protocol Datagrams over Point-to-Point Links</i>. It is a user level daemon that works in concert with the IP-Dialup driver (ipdcm) and PPP streams module (ppp(7M)) to provide IP network services over an analog modem using dialed voice grade telephone lines. The link manager automates the process of connecting to a peer (remote) host when PPP service with that host is required. The connection process can be initiated either by sending an IP datagram to a (disconnected) peer host or by receiving a notification that a peer host desires to establish a connection.</p> <p>aspppls is the <i>login service</i> that connects the peer host machine to aspppd. aspppls is invoked by the serial port monitor when a peer machine logs into a PPP enabled account. Its purpose is to cause the link manager to accept the incoming call.</p>
OPTIONS	<p>-d <i>debug-level</i> The <i>debug-level</i> is a number between 0 and 9. Higher numbers give more detailed debugging information. The output is written to the log file <code>/etc/log/asppp.log</code>.</p>
USAGE	<p>The link manager is invoked at boot time if the configuration file <code>/etc/asppp.cf</code> is present. After parsing the configuration file and building a <i>path</i> object for each peer host, it sleeps until (1) an IP datagram is routed to one of the ipd or ipdptp interfaces (see ppp(7M)), or (2) it is notified by the login service that a peer host is attempting to make a connection.</p> <p>In the first case, it consults the UUCP database, dials the modem, logs into the peer host, establishes the PPP data link, brings up IP, and forwards the IP datagram that initiated the process.</p> <p>In the second case, the link manager opens the file descriptor supplied by the login service, establishes the PPP data link, and brings up IP.</p> <p>If the link manager determines that there has been no IP traffic for the period specified by the inactivity_timeout keyword, it disconnects the link by bringing down IP and PPP and closing the connection with the peer host.</p> <p>The link manager can be reinitialized by sending it the -HUP signal (with kill(1) for example), which causes it to disconnect all open PPP links and reread the configuration file.</p>
Path	<p>A <i>path</i> is an object that contains the state of a connection with a peer host. Information such as system names, interface names, timeout values, and other attributes are kept in the path object. There exists a path for each potential peer host. Paths are defined in the</p>

configuration file.	
Interfaces	<p>The link manager supports two types of IP layer interfaces; the point-to-multipoint interface (ipd) and the point-to-point interface (ipdptp) (see ppp(7M)).</p> <p>The point-to-multipoint interface logically connects the host machine to a network containing one or more peer hosts. IP traffic to or from any of the peer hosts is routed through the point-to-multipoint interface. When an ipd interface is configured, only one IP address, that of the host, is assigned. In other words, it behaves very similarly to an Ethernet interface, although the broadcast capability is not supported. This type of interface is well suited for a dial in PPP server.</p> <p>The point-to-point interface logically connects the host machine with one peer host. Only IP traffic to or from the peer host is routed through this interface. When an ipdptp interface is configured, two IP addresses are assigned. This type of interface is well suited to support a remote, or nomadic, machine.</p> <p>An interface must be fully configured and enabled (i.e. up) before an IP datagram will be routed to it. It's also true that a point-to-multipoint interface must be fully configured and enabled before the link manager will associate an incoming connection with it. It's not necessary, however, for a point-to-point interface to be configured and enabled before an incoming connection will be assigned to it. A point-to-point interface that is "plumbed", but otherwise not configured or enabled (i.e. down), can be used to accept an incoming connection if the path associated with the potential connection contains a dynamic interface specification (e.g. interface ipdptp*). In this case the link manager will select a disabled (down) interface, configure the host and peer addresses, bring it up, and assign it for the duration of the connection.</p>
Routing	<p>Special attention should be paid to routing issues that may arise if a host has more than one interface configured and enabled. By definition, a host with more than one enabled interface is a <i>router</i>, and the routing daemon (typically in.routed) will advertise the routes provided by the PPP interfaces. This is normally acceptable behavior for a dial in server, but can cause network disruptions if not administered properly.</p> <p>To prevent routing information packets (RIP) from flowing over point-to-point interfaces, specify the norip keyword followed by the interface name in the /etc/gateways file. These entries, for example, prevent RIP from being sent over ipdptp0 and ipdptp1:</p> <pre style="margin-left: 40px;">norip ipdptp0 norip ipdptp1</pre> <p>See in.routed(1M) for further information.</p>
Authentication	<p>The link manager can be configured to support either the Password Authentication Protocol (PAP) or the Challenge Handshake Authentication Protocol (CHAP) as specified in RFC1334. Both protocols can be configured simultaneously, in which case, CHAP has precedence. A single host may participate as an authenticator (the local host requests that the peer host authenticate itself) or an authenticatee (the local host has been asked by the peer host to authenticate itself) or as both. It is also possible for a host to be an authenticator for one protocol and an authenticatee for the other protocol.</p>

PAP is a simple protocol similar to a standard login/password type of authentication. The PAP authenticator sends a message to its peer requesting that the peer authenticate itself. The peer responds with an authenticate request packet that contains an id and a password (both in plaintext). The id and password are matched against a local copy, and if they match, the connection is established. If they don't match, the connection is dropped.

CHAP does not pass any plaintext authentication data across the link. The CHAP authenticator sends a challenge packet to the peer that contains a random string. The peer then takes the string in the challenge packet and computes a response string that is a function of the challenge string and a shared secret key. The peer then sends a response packet back to the authenticator. The authenticator computes a string based on the original challenge string and the shared secret key and matches that result with the received response. If they match, the connection is established. Otherwise the connection is dropped.

Configuration File

The primary purpose of the `/etc/asppp.cf` configuration file is to define each path used by the link manager to establish and maintain communication with a peer system.

The file consists of a sequence of tokens separated by white space (blanks, tabs, and new lines). There are no record boundaries or any other constraints on the placement of the tokens. If a token begins with a pound sign (#), all characters between the pound sign and the next newline (\n) are ignored (i.e. they are treated as a comment). Alphanumeric tokens are case insensitive and are translated by the lexical analyzer into lower case before further processing.

A *string* is a single token that does not contain embedded white space. The standard ANSI C \ escape sequence may be used to embed special characters (see an ANSI C manual for a list of escaped special characters). Use \s for the space character. If a pound sign appears at the beginning of a *string*, it must be escaped (\#) to avoid interpretation as a comment. A NULL (\0) will truncate the *string*.

Groups of tokens are assembled into units known as *paths* (essentially a human-readable form of the path object). A path begins with the keyword **path** and ends at the token found before any subsequent **path** (or **defaults**) keyword or at the last token in the file. The tokens comprising a path are further partitioned into small groups consisting mostly of keyword/value pairs that define the attributes of the current path. If a particular keyword/value pair is not listed for a path, the default value is assumed.

The token sequences that begin with the substrings **ipcp_** or **lcp_** refer to PPP initial configuration options as specified in RFC1332, *The PPP Internet Protocol Control Protocol (IPCP)*. See the RFC for a more complete definition of these options.

The following is an alphabetic list of the token sequences that can be contained in a configuration file. Required sequences are noted.

Keywords

chap_name *string*

One or more octets representing the identification of this host. The name should not be NUL or CR/LF terminated. The name is sent to the authenticator in a response packet. Place this key/value pair in the authenticatee's

configuration file.

chap_peer_secret *string*

One or more octets, preferably at least sixteen, that contain the secret key that is used with the challenge value to generate the string to match with the response received from the peer. Place this key/value pair in the authenticator's configuration file.

chap_peer_name *string*

One or more octets representing the identification of the peer transmitting the packet. The name should not be NUL or CR/LF terminated. The name is received from the peer in a response packet. Place this key/value pair in the authenticator's configuration file.

chap_secret *string*

One or more octets, preferably at least sixteen, that contain the secret key that is used with the received challenge value to generate the response sent to the authenticator. Place this key/value pair in the authenticatee's configuration file.

debug_level *number*

number is between 0 and 9. Higher numbers give more detailed debugging information as shown in the table below. The output is written to the `/etc/log/asppp.log` file. The value set by the **debug_level** keyword overrides the **-d** command line option.

<u>level</u>	<u>meaning</u>
0	errors only
1	minimal information
4	some uucp chat-script info
5	all uucp chat-script info
7	maximum uucp info
8	PPP message traces
9	Raw IP packets

defaults Indicates that all following token sequences up the next **path** keyword, or the end of file, set default attributes that affect subsequently defined paths.

default_route

When the IP layer corresponding to the current path is fully operational, add the peer IP address to the route table as the default destination. The route is removed when the IP layer is brought down. Note: the **default_route** keyword is only installed by point-to-point interfaces.

ifconfig *parameters*

(Required) The **ifconfig** keyword and associated *parameters* are passed to the shell for evaluation and execution. It's used to define an interface. See the **ifconfig(1M)** man page for more information.

inactivity_timeout *seconds*

seconds is the maximum number of seconds that the connection associated

with the current path can remain idle before it is terminated. **0** may be specified to indicate no timeout. The default is 120 seconds.

interface (ipdn | ipdptn | ipdptp*)

(Required) Associates a specific point-to-multipoint or point-to-point interface as denoted by the non-negative integer *n* with the current path. The third form, **ipdptp***, indicates that the interface associated with the path is a dynamic interface that will be selected at connect time from a pool of previously configured, inactive (down) point-to-point interfaces.

ipcp_async_map hex-number

Specifies the async control character map for the current path. The *hex-number* is the natural (i.e. big endian) form representation of the four octets that comprise the map. The default value is ffffffff.

ipcp_compression (vj | off)

Indicates whether IP compression is enabled or not. If enabled (**vj**), the Van Jacobson compression algorithm is used. The default is compression (**vj**).

lcp_compression (on | off)

Indicates whether PPP address, control, and protocol field compression is enabled or not. If enabled, both the address and control field compression and the protocol field compression options are set. The default is compression (**on**).

lcp_mru number

number specifies a desired maximum receive unit packet size in octets. The default is 1500.

negotiate_address (on | off)

Indicates whether or not local IP address assignment is obtained through negotiation and assigned dynamically. If enabled, the local address will be obtained from the remote end of the PPP link. If so obtained, any local address other than 0.0.0.0 can be used to initially configure the interface. The default is to negotiate (**on**).

pap_id string

One or more octets that represent the name of the host which will be sent to the authenticator. To indicate a zero length string, do not include the keyword. Place this key/value pair in the authenticatee's configuration file.

pap_password string

One or more octets that indicate the password for this host which will be sent to the authenticator. To indicate a zero length string, do not include the keyword. Place this key/value pair in the authenticatee's configuration file.

pap_peer_id string

One or more octets that indicate the name of the peer to be authenticated. To indicate a zero length string, do not include the keyword. Place this key/value pair in the authenticator's configuration file.

pap_peer_password string

One or more octets that indicate the password to be used for authentication. To indicate a zero length string, do not include the keyword. Place this key/value pair in the authenticator's configuration file.

path (Required) Indicates that all following token sequences are to be grouped together as attributes of this (current) path. The collection of attributes comprising the current path are terminated by the occurrence of a subsequent **path** or **defaults** keyword or by the end of file.

peer_ip_address *IP-address*

(Required for point-to-multipoint paths) Associates the *IP-address* with the current path. The value is ignored if the path specifies a point-to-point interface. The *IP-address* may be in "dotted decimal", hexadecimal, or symbolic (i.e. hostname) format.

peer_system_name *name*

(Required) Associates the peer system *name* with the current path. The name is used to look up modem and peer specific information for outbound connections in the UUCP `/etc/uucp/Systems` file. For incoming connections, the appropriate path is determined by matching *name* with the login name that was used to obtain the connection (i.e. an entry in the `/etc/passwd` file specifies *name* in the username field).

require_authentication (*off* | **pap** [**chap**] | **chap** [**pap**])

Indicates that the local host is the authenticator, and that the peer is required to authenticate itself. If either **pap** or **chap** is present, the peer must participate in the authentication protocol or the connection will be terminated. If both **pap** and **chap** are present, then the local host will first try to negotiate **chap**, and if that fails, it will try to negotiate **pap**. If both fail, the connection will be terminated. The default does not require authentication (*off*).

If **pap** is required, then the **pap_peer_id** and **pap_peer_password** keywords and values should be specified for the associated path. If they are not specified, the corresponding values are set to the NULL string. If **chap** is required then the **chap_peer_name** and **chap_peer_secret** keywords and values must be specified for the associated path.

version *n* Specifies that the contents of the configuration file correspond to format version *n*. If this keyword is present, it must be the first keyword in the file. If absent, the version is assumed to be 1. This document contains the definition of the version 1 format for the configuration file.

will_do_authentication (*off* | **pap** [**chap**] | **chap** [**pap**])

Indicates that the local host is a potential authenticatee and is willing to participate in the specified authentication protocol. If both **pap** and **chap** are present then the local host is willing to participate in either authentication protocol. The default does not participate in authentication (*off*).

If **pap** is available, then the **pap_id** and **pap_password** keywords and values should be specified for the associated path. If they are not specified, the corresponding values are set to the NULL string. If **chap** is available then the

chap_name and **chap_secret** keywords and values must be specified for the associated path.

EXAMPLES

The following are examples of typical configuration files.

Remote Machine

In this example, the remote machine is most likely a nomadic or home machine with a single modem.

```
#
# Dial in to two servers
#
ifconfig ipdptp0 plumb nomad1 dialin1 private up
path
    interface ipdptp0
    peer_system_name Pdialin1
    will_do_authentication pap
    pap_id nomad1
    pap_password secret

ifconfig ipdptp1 plumb nomad1 dialin2 private up
path
    interface ipdptp1
    peer_system_name Pdialin2
    lcp_mru 1006
```

Dial In Servers

This example shows a dial in server supporting a point-to-multipoint interface. There may be several modems attached to this server. Note that the network addressed by the **ipd** interface will be advertised by the router, and all traffic destined for that network will be routed through this host. For that reason, it is not wise to support multiple dial in servers with point-to-multipoint interfaces to the same network.

```
#
# A point-to-multipoint dial in server
#
ifconfig ipd0 plumb dialin1 netmask + up
defaults
    interface ipd0
    inactivity_timeout 900 # 15 minutes
    require_authentication chap pap
    chap_peer_name nomads
path
    peer_system_name Pnomad1
    pap_peer_id nomad1
    pap_peer_password secret
path
    peer_system_name Pnomad2
    chap_peer_secret a\sspace
```

```

path
  peer_system_name Pnomad3
  inactivity_timeout 0           # No timeout for this host
  chap_peer_secret \#123;
path
  peer_system_name Pnomad4
  chap_peer_secret My\sSecret#Word

```

This is another dial in server that supports dynamic point-to-point interfaces. Usually the server has one modem for each interface. One advantage of using dynamic interfaces is that (host) routes will only be advertised when an interface is up. Therefore, multiple dial in servers can be supported.

```

#
# A dynamic point-to-point dial in server
#
ifconfig ipdptp0 plumb dialin2 down
ifconfig ipdptp1 plumb dialin2 down
ifconfig ipdptp2 plumb dialin2 down
defaults
  interface ipdptp*
  inactivity_timeout 900
  debug_level 5

path peer_system_name Pnomad1
path
  peer_system_name Pnomad2
  negotiate_address off
path peer_system_name Pnomad3
path peer_system_name Pnomad4

```

FILES	/etc/asppp.cf	configuration file
	/etc/log/asppp.log	message log file
	/etc/uucp/Devices	
	/etc/uucp/Dialers	
	/etc/uucp/Sysfiles	
	/etc/uucp/Systems	
	/tmp/.asppp.fifo	communication path between aspppd and aspppls
	/usr/sbin/aspppd	link manager
	/usr/sbin/aspppls	login service

SEE ALSO **ifconfig(1M)**, **in.routed(1M)**, **ppp(7M)**
TCP/IP and Data Communications Guide

NAME	audit – control the behavior of the audit daemon
SYNOPSIS	audit -n -s -t
AVAILABILITY	The functionality described in this man page is available only if the Basic Security Module (BSM) has been enabled. See bsmconv(1M) for more information.
DESCRIPTION	The audit command is the general administrator's interface to maintaining the audit trail. The audit daemon may be notified to read the contents of the audit_control(4) file and re-initialize the current audit directory to the first directory listed in the audit_control file or to open a new audit file in the current audit directory specified in the audit_control file as last read by the audit daemon. The audit daemon may also be signaled to close the audit trail and disable auditing.
OPTIONS	-n Signal audit daemon to close the current audit file and open a new audit file in the current audit directory. -s Signal audit daemon to read audit control file. The audit daemon stores the information internally. -t Signal audit daemon to close the current audit trail file, disable auditing and die.
DIAGNOSTICS	The audit command will exit with 0 upon success and a positive integer upon failure.
FILES	/etc/security/audit_user /etc/security/audit_control
SEE ALSO	bsmconv(1M) , praudit(1M) , audit(2) , audit_control(4) , audit_user(4)
NOTES	This command does not modify a process's preselection mask. It only affects which audit directories are used for audit data storage and to specify the minimum size free.

NAME	audit_startup – audit subsystem initialization script
SYNOPSIS	/etc/security/audit_startup
AVAILABILITY	The functionality described in this man page is available only if the Basic Security Module (BSM) has been enabled. See bsmconv(1M) for more information.
DESCRIPTION	The audit_startup script is used to initialize the audit subsystem before the audit daemon is started. This script is configurable by the system administrator, and currently consists of a series of auditconfig(1M) commands to set the system default policy, and download the initial event to class mapping.
SEE ALSO	auditconfig(1M) , auditd(1M) , bsmconv(1M)

NAME	audit_warn – audit daemon warning script
SYNOPSIS	<code>/etc/security/audit_warn [option [arguments]]</code>
AVAILABILITY	The functionality described in this man page is available only if the Basic Security Module (BSM) has been enabled. See bsmconv(1M) for more information.
DESCRIPTION	<p>The audit_warn script processes warning or error messages from the audit daemon. When a problem is encountered, the audit daemon, auditd(1M) calls audit_warn with the appropriate arguments. The <i>option</i> argument specifies the error type.</p> <p>The system administrator can specify a list of mail recipients to be notified when an audit_warn situation arises by defining a mail alias called <i>audit_warn</i> in aliases(4). The users that make up the <i>audit_warn</i> alias are typically the audit and root users.</p>
OPTIONS	<p>allhard <i>count</i> Indicates that the hard limit for all filesystems has been exceeded <i>count</i> times. The default action for this option is to send mail to the <i>audit_warn</i> alias only if the <i>count</i> is 1, and to write a message to the machine console every time. It is recommended that mail <i>not</i> be sent every time as this could result in a the saturation of the file system that contains the mail spool directory.</p> <p>allsoft Indicates that the soft limit for all filesystems has been exceeded. The default action for this option is to send mail to the <i>audit_warn</i> alias and to write a message to the machine console.</p> <p>auditoff Indicates that someone other than the audit daemon changed the system audit state to something other than AUC_AUDITING. The audit daemon will have exited in this case. The default action for this option is to send mail to the <i>audit_warn</i> alias and to write a message to the machine console.</p> <p>ebusy Indicates that the audit daemon is already running. The default action for this option is to send mail to the <i>audit_warn</i> alias and to write a message to the machine console.</p> <p>getacdir <i>count</i> Indicates that there is a problem getting the directory list from audit_control(4). The audit daemon will hang in a sleep loop until the file is fixed. The default action for this option is to send mail to the <i>audit_warn</i> alias only if <i>count</i> is 1, and to write a message to the machine console every time. It is recommended that mail <i>not</i> be sent every time as this could result in a the saturation of the file system that contains the mail spool directory.</p>

hard filename

Indicates that the hard limit for the file has been exceeded. The default action for this option is to send mail to the *audit_warn* alias and to write a message to the machine console.

nostart

Indicates that auditing could not be started. The default action for this option is to send mail to the *audit_warn* alias and to write a message to the machine console. Some administrators may prefer to modify *audit_warn* to reboot the system when this error occurs.

postsigterm

Indicates that an error occurred during the orderly shutdown of the audit daemon. The default action for this option is to send mail to the *audit_warn* alias and to write a message to the machine console.

soft filename

Indicates that the soft limit for *filename* has been exceeded. The default action for this option is to send mail to the *audit_warn* alias and to write a message to the machine console.

tmpfile

Indicates that the temporary audit file already exists indicating a fatal error. The default action for this option is to send mail to the *audit_warn* alias and to write a message to the machine console.

SEE ALSO

audit(1M), auditd(1M), bsmconv(1M), aliases(4), audit.log(4), audit_control(4)

NAME	auditconfig – configure auditing
SYNOPSIS	auditconfig [<i>args</i>]
AVAILABILITY	The functionality described in this man page is available only if the Basic Security Module (BSM) has been enabled. See bsmconv (1M) for more information.
DESCRIPTION	auditconfig provides a command line interface to get and set kernel audit parameters.
OPTIONS	<p>-chkconf Check the configuration of kernel audit event to class mappings. If the runtime class mask of a kernel audit event does not match the configured class mask, a mismatch is reported.</p> <p>-conf Configure kernel audit event to class mappings. Runtime class mappings are changed to match those in the audit event to class database file.</p> <p>-getcond Display the kernel audit condition. The condition displayed is the literal string auditing meaning auditing is enabled and turned on (the kernel audit module is constructing and queuing audit records) or noaudit meaning auditing is enabled but turned (the kernel audit module is not constructing and queuing audit records), or disabled meaning that the audit module has not been enabled. See auditon(2) and auditd(1M) for further information.</p> <p>-setcond[<i>auditing</i> <i>noaudit</i>] Set the kernel audit condition to the <i>condition</i> specified where <i>condition</i> is the literal string auditing indicating auditing should be enabled or noaudit indicating auditing should be disabled.</p> <p>-getclass <i>event</i> Display the preselection mask associated with the specified kernel audit event. <i>event</i> is the kernel event number or event name.</p> <p>-setclass <i>event</i> <i>audit_flag</i>,<i>audit_flag</i> . . .] Map the kernel event <i>event</i> to the classes specified by <i>audit_flags</i>. <i>event</i> is an event number or name. An <i>audit_flag</i> is a two character string representing an audit class. See audit_control(4) for further information.</p> <p>-lsevent Display the currently configured (runtime) kernel and user level audit event information.</p> <p>-getpinfo <i>pid</i> Display the audit ID, preselection mask, terminal ID and audit session ID for the specified process.</p> <p>-setpmask <i>pid</i> <i>flags</i> Set the preselection mask of the specified process. flags is the ASCII representation of the flags similar to that in audit_control(4).</p> <p>-setsmask <i>asid</i> <i>flags</i> Set the preselection mask of all processes with the specified audit session ID.</p>

-setumask *audit flags*
Set the preselection mask of all processes with the specified audit ID.

-lspolicy Display the kernel audit policies with a description of each policy.

-getpolicy Display the kernel audit policy.

-setpolicyf1[+ / -]*policy_flag*[,*policy_flag* ...]
Set the kernel audit policy. A policy *policy_flag* is literal strings that denotes an audit policy. A prefix of + adds the policies specified to the current audit policies. A prefix of - removes the policies specified from the current audit policies. The following are the valid policy flag strings (auditconfig -lspolicy also lists the current valid audit policy flag strings):

arge Include the **execv(2)** system call environment arguments to the audit record. This information is not included by default.

argv Include the **execv(2)** system call parameter arguments to the audit record. This information is not included by default.

cnt Do not suspend processes when audit resources are exhausted. Instead, drop audit records and keep a count of the number of records dropped. By default, process are suspended until audit resources become available.

group Include the supplementary group token in audit records. By default, the group token is not included.

path Add secondary path tokens to audit record. These are typically the pathnames of dynamically linked shared libraries or command interpreters for shell scripts. By default, they are not included.

trail Include the trailer token in every audit record. By default, the trailer token is not included.

seq Include the sequence token as part of every audit record. By default, the sequence token is not included. The sequence token attaches a sequence number to every audit record.

EXAMPLES

```
#
# map kernel audit event number 10 to the "fr" audit class
#
% auditconfig -setclass 10 fr

#
# turn on inclusion of exec arguments in exec audit records
#
% auditconfig -setpolicy +argv
```

ERRORS **auditconfig** returns **0** upon success and **1** upon failure.

FILES **/etc/security/audit_event**
/etc/security/audit_class

SEE ALSO **auditd(1M)**, **bsmconv(1M)**, **praudit(1M)**, **auditon(2)**, **execv(2)**, **audit_class(4)**,
audit_control(4), **audit_event(4)**

NAME	auditd – audit daemon
SYNOPSIS	<code>/usr/sbin/auditd</code>
AVAILABILITY	The functionality described in this man page is available only if the Basic Security Module (BSM) has been enabled. See bsmconv(1M) for more information.
DESCRIPTION	<p>The audit daemon controls the generation and location of audit trail files. If auditing is desired, auditd reads the audit_control(4) file to get a list of directories into which audit files can be written and the percentage limit for how much space to reserve on each filesystem before changing to the next directory.</p> <p>If auditd receives the signal SIGUSR1, the current audit file is closed and another is opened. If SIGHUP is received, the current audit trail is closed, the audit_control file reread, and a new trail is opened. If SIGTERM is received, the audit trail is closed and auditing is terminated. The program audit(1M) sends these signals and is recommended for this purpose.</p> <p>Each time the audit daemon opens a new audit trail file, it updates the file audit_data(4) to include the correct name.</p>
Auditing Conditions	<p>The audit daemon invokes the program audit_warn(1M) under the following conditions with the indicated options:</p> <p>audit_warn soft <i>pathname</i> The file system upon which <i>pathname</i> resides has exceeded the minimum free space limit defined in audit_control(4). A new audit trail has been opened on another file system.</p> <p>audit_warn allsoft All available file systems have been filled beyond the minimum free space limit. A new audit trail has been opened anyway.</p> <p>audit_warn hard <i>pathname</i> The file system upon which <i>pathname</i> resides has filled or for some reason become unavailable. A new audit trail has been opened on another file system.</p> <p>audit_warn allhard <i>count</i> All available file systems have been filled or for some reason become unavailable. The audit daemon will repeat this call to audit_warn every twenty seconds until space becomes available. <i>count</i> is the number of times that audit_warn has been called since the problem arose.</p> <p>audit_warn ebusy There is already an audit daemon running.</p> <p>audit_warn tmpfile The file <code>/etc/security/audit/audit_tmp</code> exists, indicating a fatal error.</p>

audit_warn nostart

The internal system audit condition is AUC_FCHDONE. Auditing cannot be started without rebooting the system.

audit_warn auditoff

The internal system audit condition has been changed to not be AUC_AUDITING by someone other than the audit daemon. This causes the audit daemon to exit.

audit_warn postsigterm

An error occurred during the orderly shutdown of the auditing system.

audit_warn getacdir

There is a problem getting the directory list from **/etc/security/audit/audit_control**.

The audit daemon will hang in a sleep loop until this file is fixed.

FILES

/etc/security/audit/audit_control

/etc/security/audit/audit_data

SEE ALSO

audit(1M), audit_warn(1M), bsmconv(1M), praudit(1M), auditon(2), auditsvc(2), audit.log(4), audit_control(4), audit_data(4)

NAME	auditreduce – merge and select audit records from audit trail files
SYNOPSIS	auditreduce [<i>options</i>] [<i>audit-trail-file</i> . . .]
AVAILABILITY	The functionality described in this man page is available only if the Basic Security Module (BSM) has been enabled. See bsmconv (1M) for more information.
DESCRIPTION	<p>auditreduce allows you to select or merge records from audit trail files. Audit files may be from one or more machines.</p> <p>The merge function merges together audit records from one or more input audit trail files into a single output file. The records in an audit trail file are assumed to be sorted in chronological order (oldest first) and this order is maintained by auditreduce in the output file.</p> <p>Unless instructed otherwise, auditreduce will merge the entire audit trail, which consists of all the audit trail files in the directory structure <i>audit_root_dir</i>/*/files (see audit_control(4) for details of the structure of the audit root). Unless stated with the -R or -S option, <i>audit_root_dir</i> defaults to /etc/security/audit. By using the file selection options it is possible to select some subset of these files, or files from another directory, or files named explicitly on the command line.</p> <p>The select function allows audit records to be selected on the basis of numerous criteria relating to the record's content (see audit.log(4) for details of record content). A record must meet all of the <i>record-selection-option</i> criteria to be selected.</p>
Audit Trail Filename Format	<p>Any audit trail file not named on the command line must conform to the audit trail filename format. Files produced by the audit system already have this format. Output file names produced by auditreduce are in this format. It is:</p> <p style="text-align: center;"><i>start-time.end-time.suffix</i></p> <p>where <i>start-time</i> is the 14-character timestamp of when the file was opened, <i>end-time</i> is the 14-character timestamp of when the file was closed, and <i>suffix</i> is the name of the machine which generated the audit trail file, or some other meaningful suffix (e.g., all, if the file contains a combined group of records from many machines). The <i>end-time</i> may be the literal string not_terminated, to indicate that the file is still being written to by the audit system. Timestamps are of the form <i>yyyymmddhhmmss</i> (year, month, day, hour, minute, second). The timestamps are in Greenwich Mean Time (GMT).</p>
OPTIONS File Selection Options	<p>The file selection options indicate which files are to be processed and certain types of special treatment.</p> <p>-A All of the records from the input files will be selected regardless of their timestamp. This option effectively disables the -a, -b, and -d options. This is useful in preventing the loss of records if the -D option is used to delete the input files after they are processed. Note, however, that if a record is <i>not</i> selected due to another option, then -A will not override that.</p>

- C** Only process complete files. Files whose filename *end-time* timestamp is **not_terminated** are not processed (such a file is currently being written to by the audit system). This is useful in preventing the loss of records if **-D** is used to delete the input files after they are processed. It does not apply to files specified on the command line.
- D suffix**
Delete input files after they are processed. The files are only deleted if the entire run is successful. If **auditreduce** detects an error while reading a file, then that file is not deleted. If **-D** is specified, **-A**, **-C** and **-O** are also implied. *suffix* is given to the **-O** option. This helps prevent the loss of audit records by ensuring that all of the records are written, only complete files are processed, and the records are written to a file before being deleted. Note that if both **-D** and **-O** are specified in the command line, the order of specification is significant. The *suffix* associated with the latter specification is in effect.
- M machine**
Allows selection of records from files with *machine* as the filename suffix. If **-M** is not specified, all files are processed regardless of suffix. **-M** can also be used to allow selection of records from files that contain combined records from many machines and have a common suffix (such as **all**).
- O suffix**
Direct output stream to a file in the current **audit_root_dir** with the indicated suffix. *suffix* may alternatively contain a full pathname, in which case the last component is taken as the suffix, ahead of which the timestamps will be placed, ahead of which the remainder of the pathname will be placed.

If the **-O** option is not specified, the output is sent to the standard output. When **auditreduce** places timestamps in the filename, it uses the times of the first and last records in the merge as the *start-time* and *end-time*.
- Q** Quiet. Suppress notification about errors with input files.
- R pathname**
Specify the pathname of an alternate audit root directory *audit_root_dir* to be *pathname*. Therefore, rather than using **/etc/security/audit/***/files by default, *pathname/**/files will be examined instead.
- S server**
This option causes **auditreduce** to read audit trail files from a specific location (server directory). *server* is normally interpreted as the name of a subdirectory of the audit root, therefore **auditreduce** will look in *audit_root_dir/server*/files for the audit trail files.

But if *server* contains any '/' characters, it is the name of a specific directory not necessarily contained in the audit root. In this case, *server*/files will be consulted.

This option allows archived files to be manipulated easily, without requiring that they be physically located in a directory structure like that of `/etc/security/audit`.

- V** Verbose. Display the name of each file as it is opened, and how many records total were written to the output stream.

Record Selection Options

The record selection options listed below are used to indicate which records are written to the output file produced by **auditreduce**.

Note: Multiple arguments of the same type are not permitted.

-a *date-time*

Select records that occurred at or after *date-time*. The *date-time* argument is described under **Option Arguments**, below. *date-time* is in local time. The **-a** and **-b** options can be used together to form a range.

-b *date-time*

Select records that occurred before *date-time*.

-c *audit-classes*

Select records by audit class. Records with events that are mapped to the audit classes specified by *audit-classes* are selected. Audit class names are defined in **audit_class(4)**. The *audit-classes* can be a comma separated list of *audit flags* like those described in **audit_control(4)**. Using the *audit flags*, one can select records based upon success and failure criteria.

-d *date-time*

Select records that occurred on a specific day (a 24-hour period beginning at 00:00:00 of the day specified and ending at 23:59:59). The day specified is in local time. The time portion of the argument, if supplied, is ignored. Any records with timestamps during that day are selected. If any hours, minutes, or seconds are given in *time*, they are ignored. **-d** can not be used with **-a** or **-b**.

-e *effective-user*

Select records with the specified *effective-user*.

-f *effective-group*

Select records with the specified *effective-group*.

-g *real-group*

Select records with the specified *real-group*.

-j *subject-ID*

Select records with the specified *subject-ID* where *subject-ID* is a process ID.

-m *event*

Select records with the indicated *event*. The *event* is the literal string or the *event* number.

-o *object_type=objectID_value*

Select records by object type. A match occurs when the record contains the information describing the specified *object_type* and the object ID equals the value specified by *objectID_value*. The allowable object types and values are as follows:

file=*pathname*

Select records containing file system objects with the specified *pathname*, where *pathname* is a comma separated list of regular expressions. If a regular expression is preceeded by a tilde (~), files matching the expression are excluded from the output. For example, the option `file="/usr/openwin,/usr,/etc"` would select all files in **/usr** or **/etc** except those in **/usr/openwin**. The order of the regular expressions is important because `auditreduce` processes them from left to right, and stops when a file is known to be either selected or excluded. Thus the option `file=/usr,/etc,~/usr/openwin` would select all files in **/usr** and all files in **/etc**. Files in **/usr/openwin** are not excluded because the regular expression **/usr** is matched first. Care should be given in surrounding the *pathname* with quotes so as to prevent the shell from expanding any tildas.

msgqid=ID

Select records containing message queue objects with the specified *ID* where *ID* is a message queue ID.

pid=ID Select records containing process objects with the specified *ID* where *ID* is a process ID. Note: Process are objects when they are receivers of signals.

semid=ID

Select records containing semaphore objects with the specified *ID* where *ID* is a semaphore ID.

shmid=ID

Select records containing shared memory objects with the specified *ID* where *ID* is a shared memory ID.

sock=port_number | machine

Select records containing socket objects with the specified *port_number* or the specified *machine* where *machine* is a machine name as defined in **hosts(4)**.

-r real-user

Select records with the specified *real-user*.

-u audit-user

Select records with the specified *audit-user*.

When one or more *filename* arguments appear on the command line, only the named files are processed. Files specified in this way need not conform to the audit trail filename format. However, **-M**, **-S**, and **-R** may not be used when processing named files. If the *filename* is "-" then the input is taken from the standard input.

Option Arguments**audit-trail-file**

An audit trail file as defined in **audit.log(4)**. An audit trail file not named on the command line must conform to the audit trail file name format. Audit trail files produced as output of **auditreduce** are in this format as well. The format is:

start-time . end-time . suffix

start-time is the 14 character time stamp denoting when the file was opened. *end-time* is the 14 character time stamp denoting when the file was closed. *end-time* may also be the literal string **not_terminated**, indicating the file is still be written to by the audit daemon or the file was not closed properly (a system crash or abrupt halt occurred). *suffix* is the name of the machine that generated the audit trail file (or some other meaningful suffix; e.g. **all** would be a good suffix if the audit trail file contains a combined group of records from many machines).

date-time

The *date-time* argument to **-a**, **-b**, and **-d** can be of two forms: An absolute *date-time* takes the form:

`yyyymmdd [hh [mm [ss]]]`

where *yyyy* specifies a year (with 1970 as the earliest value), *mm* is the month (01-12), *dd* is the day (01-31), *hh* is the hour (00-23), *mm* is the minute (00-59), and *ss* is the second (00-59). The default is 00 for *hh*, *mm* and *ss*.

An offset can be specified as: **+nd | h | m | s** where *n* is a number of units, and the tags **d**, **h**, **m**, and **s** stand for days, hours, minutes and seconds, respectively. An offset is relative to the starting time. Thus, this form can only be used with the **-b** option.

event The literal string or ordinal event number as found in **audit_event(4)**. If *event* is not found in the **audit_event** file it is considered invalid.

group The literal string or ordinal group ID number as found in **group(4)**. If *group* is not found in the **group** file it is considered invalid. *group* may be negative.

pathname

A regular expression describing a pathname.

user The literal username or ordinal user ID number as found in **passwd(4)**. If the username is not found in the **passwd** file it is considered invalid. *user* may be negative.

EXAMPLES

praudit(1M) is available to display audit records in a human-readable form.

This will display the entire audit trail in a human-readable form:

`% auditreduce | praudit`

If all the audit trail files are being combined into one large file, then deleting the original files could be desirable to prevent the records from appearing twice:

`% auditreduce -V -D /etc/security/audit/combined/all`

This will print what user **milner** did on April 13, 1988. The output will be displayed in a human-readable form to the standard output:

`% auditreduce -d 19880413 -u milner | praudit`

The above example may produce a large volume of data if **milner** has been busy.

Perhaps looking at only login and logout times would be simpler. The **-c** option will select records from a specified class:

```
% auditreduce -d 19880413 -u milner -c lo | praudit
```

To see **milner**'s login/logout activity for April 13, 14, and 15 the following is used. The results are saved to a file in the current working directory. Note that the name of the output file will have **milnerlo** as the *suffix*, with the appropriate timestamp prefixes. Note that the long form of the name is used for the **-c** option:

```
% auditreduce -a 19880413 -b +3d -u milner -c login_logout -O milnerlo
```

To follow **milner**'s movement about the file system on April 13, 14, and 15 the **chdir** record types could be viewed. Note that in order to get the same time range as the above example we needed to specify the **-b** time as the day after our range. This is because **19880416** defaults to midnight of that day, and records before that fall on **0415**, the end-day of the range.

```
% auditreduce -a 19880413 -b 19880416 -u milner -m AUE_CHDIR | praudit
```

In this example the audit records are being collected in summary form (the login/logout records only). The records are being written to a summary file in a different directory than the normal audit root to prevent the selected records from existing twice in the audit root.

```
% auditreduce -d 19880330 -c lo -O /etc/security/audit_summary/logins
```

If activity for user ID 9944 has been observed, but that user is not known to the system administrator, then the following example will search the entire audit trail for any records generated by that user. **auditreduce** will query the system as to the current validity of ID 9944, and print a warning message if it is not currently active:

```
% auditreduce -O /etc/security/audit_suspect/user9944 -u 9944
```

FILES

/etc/security/audit/server/files/* location of audit trails, when stored

SEE ALSO

bsmconv(1M), **praudit(1M)**, **audit.log(4)**, **audit_class(4)**, **audit_control(4)**, **group(4)**, **hosts(4)**, **passwd(4)**

DIAGNOSTICS

auditreduce will print out error messages if there are command line errors and then exit. If there are fatal errors during the run **auditreduce** will print an explanatory message and exit. In this case the output file may be in an inconsistent state (no trailer or partially written record) and **auditreduce** will print a warning message before exiting. Successful invocation returns **0** and unsuccessful invocation returns **1**.

Since **auditreduce** may be processing a large number of input files, it is possible that the machine-wide limit on open files will be exceeded. If this happens, **auditreduce** will print a message to that effect, give information on how many file there are, and exit.

If **auditreduce** prints a record's timestamp in a diagnostic message, that time is in local time. However, when filenames are displayed, their timestamps are in GMT.

BUGS

Conjunction, disjunction, negation, and grouping of record selection options should be allowed.

NAME	auditstat – display kernel audit statistics
SYNOPSIS	auditstat [-c <i>count</i>] [-h <i>numlines</i>] [-i <i>interval</i>] [-n] [-v]
AVAILABILITY	The functionality described in this man page is available only if the Basic Security Module (BSM) has been enabled. See bsmconv (1M) for more information.
DESCRIPTION	<p>auditstat displays kernel audit statistics. The fields displayed are as follows:</p> <p>aud The total number of audit records processed by the audit(2) system call.</p> <p>ctl This field is obsolete.</p> <p>drop The total number of audit records that have been dropped. Records are dropped according to the kernel audit policy. See auditon(2), AUDIT_CNT policy for details.</p> <p>enq The total number of audit records put on the kernel audit queue.</p> <p>gen The total number of audit records that have been constructed (not the number written).</p> <p>kern The total number of audit records produced by user processes (as a result of system calls).</p> <p>mem The total number of Kbytes of memory currently in use by the kernel audit module.</p> <p>nona The total number of non-attributable audit records that have been constructed. These are audit records that are not attributable to any particular user.</p> <p>rblk The total number of times that auditsvc(2) has blocked waiting to process audit data.</p> <p>tot The total number of Kbytes of audit data written to the audit trail.</p> <p>wblk The total number of times that user processes blocked on the audit queue at the high water mark.</p> <p>wrtn The total number of audit records written. The difference between enq and wrtn is the number of outstanding audit records on the audit queue that have not been written.</p>

OPTIONS**-c** *count*

Display the statistics a total of *count* times. If *count* is equal to zero, statistics are displayed indefinitely. A time interval must be specified.

-h *numlines*

Display a header for every *numlines* of statistics printed. The default is to display the header every 20 lines. If *numlines* is equal to zero, the header is never displayed.

-i *interval*

Display the statistics every *interval* where *interval* is the number of seconds to sleep between each collection.

-n

Display the number of kernel audit events currently configured.

-v

Display the version number of the kernel audit module software.

ERRORS

auditstat returns 0 upon success and 1 upon failure.

SEE ALSO

auditconfig(1M), **praudit(1M)**, **bsmconv(1M)**, **audit(2)**, **auditon(2)**, **auditsvc(2)**

NAME	automount – install automatic mount points
SYNOPSIS	<code>/usr/sbin/automount [-t <i>duration</i>] [-v]</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>automount is a command that installs autofs mount points and associates an automount map with each mount point. The autofs filesystem monitors attempts to access directories within it and notifies the automountd(1M) daemon. The daemon uses the map to locate a filesystem, which it then mounts at the point of reference within the autofs filesystem. You can assign a map to an autofs mount using an entry in the /etc/auto_master map or a direct map.</p> <p>If the file system is not accessed within an appropriate interval (five minutes by default), the automountd daemon unmounts the file system.</p> <p>The file /etc/auto_master determines the locations of all autofs mount points. By default, this file contains four entries:</p> <pre> # Master map for automounter # +auto_master /net -hosts -nosuid /home auto_home /xfn -xfn </pre> <p>The +auto_master entry is a reference to an external NIS or NIS+ master map. If one exists, then its entries are read as if they occurred in place of the +auto_master entry. The remaining entries in the master file specify a directory on which an autofs mount will be made followed by the automounter map to be associated with it. Optional mount options may be supplied as an optional third field in the each entry. These options are used for any entries in the map that do not specify mount options explicitly. The automount command is usually run without arguments. It compares the entries /etc/auto_master with the current list of autofs mounts in /etc/mnttab and adds, removes or updates autofs mounts to bring the /etc/mnttab up to date with the /etc/auto_master. At boot time it installs all autofs mounts from the master map. Subsequently, it may be run to install autofs mounts for new entries in the master map or an direct map, or to perform unmounts for entries that have been removed.</p>
OPTIONS	<p>-t <i>duration</i> Specify a <i>duration</i>, in seconds, that a file system is to remain mounted when not in use. The default is 5 minutes.</p> <p>-v Verbose mode. Notify of autofs mounts, unmounts or other non-essential information.</p>

USAGE
Map Entry Format

A simple map entry (mapping) takes the form:

key [*-mount-options*] *location* . . .

where **key** is the full pathname of the directory to mount when used in a direct map, or the simple name of a subdirectory in an indirect map. *mount-options* is a comma-separated list of **mount** options, and *location* specifies a file system from which the directory may be mounted. In the case of a simple NFS mount, *location* takes the form:

host:pathname

host is the name of the host from which to mount the file system (it may be omitted if the pathname refers to a local device on which the filesystem resides) and *pathname* is the pathname of the directory to mount.

**Replicated
 Filesystems**

Multiple *location* fields can be specified for replicated NFS filesystems, in which case **automount** chooses a server with preference given to a server on the local subnet or net. If each *location* in the list shares the same *pathname* then a single *location* may be used with a comma-separated list of hostnames:

hostname,hostname...:pathname

Requests for a server may be weighted, with the weighting factor appended to the server name as an integer in parentheses. Servers without a weighting are assumed to have a value of zero (most likely to be selected). Progressively higher values decrease the chance of being selected. In the example,

man -ro alpha,bravo,charlie(1),delta(4):/usr/man

hosts **alpha** and **bravo** have the highest priority; host **delta**, the lowest.

Note: Server proximity takes priority in the selection process. In the example above, if the server **delta** is on the same network segment as the client, but the others are on different network segments, then **delta** will be selected — the weighting value is ignored. The weighting has effect only when selecting between servers with the same network proximity.

In cases where each server has a different export point, you can still apply the weighting. For example:

man -ro alpha:/usr/man bravo,charlie(1):/usr/share/man delta(3):/export/man

A mapping can be continued across input lines by escaping the NEWLINE with a '\ ' (backslash). Comments begin with a '#' (number sign) and end at the subsequent NEWLINE.

Map Key Substitution

The '&' (ampersand) character is expanded to the value of the **key** field for the entry in which it occurs. In this case:

jane **sparcserver:/home/&**

the & expands to **jane**.

Wildcard Key

The '*' (asterisk) character, when supplied as the **key** field, is recognized as the catch-all entry. Such an entry will match any key not previously matched. For instance, if the following entry appeared in the indirect map for **/config**:

* **&/export/config/&**

this would allow automatic mounts in **/config** of any remote file system whose location could be specified as:

hostname:/export/config/hostname

Variable Substitution

Client specific variables can be used within an **automount** map. For instance, if **\$HOST** appeared within a map, **automount** would expand it to its current value for the client's host name. Supported variables are:

ARCH	The output of <code>uname -m</code> .	The architecture name. For example "sun4"
CPU	The output of <code>uname -p</code> .	The processor type. For example "sparc"
HOST	The output of <code>uname -n</code> .	The host name. For example "biggles"
OSNAME	The output of <code>uname -s</code> .	The OS name. For example "SunOS"
OSREL	The output of <code>uname -r</code> .	The OS release name. For example "5.3"
OSVERS	The output of <code>uname -v</code> .	The OS version. For example "beta1.0"

If a reference needs to be protected from affixed characters, you can surround the variable name with '{ }' (curly braces).

Multiple Mounts

A multiple mount entry takes the form:

key [*-mount-options*] [[*mountpoint*] [*-mount-options*] *location*...]...

The initial */[mountpoint]* is optional for the first mount and mandatory for all subsequent mounts. The optional *mountpoint* is taken as a pathname relative to the directory named by **key**. If *mountpoint* is omitted in the first occurrence, a *mountpoint* of */* (root) is implied.

Given an entry in the indirect map for **/src**:

```

beta -ro \
 / svr1,svr2:/export/src/beta \
 /1.0 svr1,svr2:/export/src/beta/1.0 \
 /1.0/man svr1,svr2:/export/src/beta/1.0/man

```

automount would automatically mount **/src/beta**, **/src/beta/1.0**, and **/src/beta/1.0/man**, as needed, from either **svr1** or **svr2**, whichever host is nearest and responds first.

Other Filesystem Types

The automounter assumes NFS mounts as a default filesystem type. Other filesystem types can be described using the **fstype** mount option. Other mount options specific to this filesystem type can be combined with the **fstype** option. The location field must contain information specific to the filesystem type. If the location field begins with a slash, a colon character must be prepended, for instance, to mount a CD filesystem:

```
cdrom -fstype=hsfs,ro :/dev/sr0
```

or to perform an **autofs** mount:

```
src -fstype=autofs auto_src
```

Mounts using CacheFS are most useful when applied to an entire map as map defaults. The following entry in the master map describes cached home directory mounts. It assumes the default location of the cache directory, **/cache**.

```
/home auto_home -fstype=cachefs,backfstype=nfs
```

Indirect Maps

An indirect map allows you to specify mappings for the subdirectories you wish to mount under the **directory** indicated on the command line. In an indirect map, each **key** consists of a simple name that refers to one or more filesystems that are to be mounted as needed.

Direct Maps

Entries in a direct map are associated directly with **autofs** mount points. Each **key** is the full pathname of an **autofs** mount point. The direct map as a whole is not associated with any single directory.

Included Maps

The contents of another map can be included within a map with an entry of the form
+mapname

If **mapname** begins with a slash then it is assumed to be the pathname of a local file. Otherwise the location of the map is determined by the policy of the name service switch according to the entry for the automounter in **/etc/nsswitch.conf**, such as

```
automount: files nis
```

If the name service is **files** then the name is assumed to be that of a local file in **/etc**. If the key being searched for is not found in the included map, the search continues with the next entry.

Special Maps

There are three special maps available: **-hosts**, **-xfs**, and **-null**. The **-hosts** map is used with the **/net** directory and assumes that the map key is the hostname of an NFS server. The **automountd** daemon dynamically constructs a map entry from the server's list of exported filesystems. For instance a reference to **/net/hermes/usr** would initiate an

automatic mount of all exported file systems from **hermes** that are mountable by the client. References to a directory under **/net/hermes** will refer to the corresponding directory relative to **hermes** root.

The **-xfn** map is used to mount the initial context of the Federated Naming Service (FNS) namespace under the **/xfn** directory. For more information on FNS, see **fns(5)**, **fns_initial_context(5)**, **fns_policies(5)**, and the Federated Naming Service Guide.

The **-null** map, when indicated on the command line, cancels a previous map for the directory indicated. This is most useful in the **/etc/auto_master** for cancelling entries that would otherwise be inherited from the **+auto_master** include entry. To be effective, the **-null** entries must be inserted before the included map entry.

Executable Maps

Local maps that have the execute bit set in their file permissions will be executed by the automounter and provided with a key to be looked up as an argument. The executable map is expected to return the content of an automounter map entry on its stdout or no output if the entry cannot be determined.

Configuration and the auto_master Map

When initiated without arguments, **automount** consults the master map for a list of **autofs** mount points and their maps. It mounts any **autofs** mounts that are not already mounted, and unmounts **autofs** mounts that have been removed from the master map or direct map.

The master map is assumed to be called **auto_master** and its location is determined by the name service switch policy. Normally the master map is located initially as a local file **/etc/auto_master**.

FILES

/etc/auto_master master automount map.
/etc/auto_home map to support automounted home directories.
/etc/nsswitch.conf the name service switch configuration file.

SEE ALSO

automountd(1M), **mount(1M)**, **fns(5)**, **fns_initial_context(5)**, **fns_policies(5)**,
NFS Administration Guide

NOTES

The **-hosts** map must mount all of the exported NFS filesystems from a server. If frequent access to just a single filesystem is required, it is more efficient to access the filesystem with a map entry that is tailored to mount just the filesystem of interest.

Autofs mount points must not be hierarchically related. **automount** does not allow an **autofs** mount point to be created within another **autofs** mount.

Since each direct map entry results in a new **autofs** mount such maps should be kept short.

If a directory contains direct map mount points then an **ls -l** in the directory will force all the direct map mounts to occur.

Entries in both direct and indirect maps can be modified at any time. The new information is used when **automountd** next uses the map entry to do a mount.

New entries added to a master map or direct map will not be useful until the automount command is run to install them as new **autofs** mount points. New entries added to an indirect map may be used immediately.

An **autofs** directory associated with an indirect map shows only currently-mounted entries. This is a deliberate policy to avoid inadvertent mounting of every entry in a map via an **ls -l** of the directory.

The multiple location feature for NFS mounts allows the **automountd** daemon to choose the most appropriate server at mount time. While such a mount is in effect, the daemon does not monitor the status of the server. If the server crashes, **automountd** will not select an alternative server from the list.

Default mount options can be assigned to an entire map when specified as an optional third field in the master map. These options apply only to map entries that have no mount options.

The Network Information Service (NIS) was formerly known as Sun Yellow Pages (YP). The functionality of the two remains the same.

NAME	automountd – autofs mount/unmount daemon
SYNOPSIS	automountd [-Tv] [-D name=value]
AVAILABILITY	SUNWcsu
DESCRIPTION	automountd is an RPC server that answers file system mount and unmount requests from the autofs filesystem. It uses local files or name service maps to locate filesystems to be mounted. These maps are described with the automount(1M) command. The automountd daemon is automatically invoked in run level 2.
OPTIONS	-T Trace. Expand each RPC call and display it on the standard output. -v Verbose. Log status messages to the console. -D name=value Assign <i>value</i> to the indicated automount map substitution variable. These assignments cannot be used to substitute variables in the master map auto_master .
FILES	/etc/auto_master master map for automounter
SEE ALSO	automount(1M)

NAME	autopush – configures lists of automatically pushed STREAMS modules
SYNOPSIS	autopush -f filename autopush -g -M major -m minor autopush -r -M major -m minor
AVAILABILITY	SUNWcsu
DESCRIPTION	This command is used to configure the list of modules to be automatically pushed onto the stream when a device is opened. It can also be used to remove a previous setting or get information on a setting.
OPTIONS	<p>-f filename Set up the autopush configuration for each driver according to the information stored in <i>filename</i>. An autopush file consists of lines of four or more fields, separated by spaces as shown below:</p> <p style="text-align: center;"><i>major minor last-minor module1 module2 . . . modulen</i></p> <p>The first field is a string that specifies the <i>major</i> device name, as listed in the /kernel/drv directory. The next two fields are integers that specify the <i>minor</i> device number and <i>last-minor</i> device number. The fields following represent the names of modules. If <i>minor</i> is -1, then all minor devices of a major driver specified by <i>major</i> are configured, and the value for <i>last-minor</i> is ignored. If <i>last-minor</i> is 0, then only a single minor device is configured. To configure a range of minor devices for a particular major, <i>minor</i> must be less than <i>last-minor</i>.</p> <p>The last fields of a line in the autopush file represent the list of module names. The maximum number of modules that can be automatically pushed on a stream is eight. The modules are pushed in the order they are specified. Comment lines start with a # sign.</p> <p>-g Get the current configuration setting of a particular <i>major</i> and <i>minor</i> device number specified with the -M and -m options respectively and displays the autopush modules associated with it. It will also return the starting minor device number if the request corresponds to a setting of a range (as described with the -f option).</p> <p>-M major Specifies the major device number.</p> <p>-m minor Specifies the minor device number.</p> <p>-r Remove the previous configuration setting of the particular <i>major</i> and <i>minor</i> device number specified with the -M and -m options respectively. If the values of <i>major</i> and <i>minor</i> correspond to a previously established setting of a range of minor devices, where <i>minor</i> matches the first minor device number in the range, the configuration would be removed for the entire range.</p>

EXAMPLES

The following example gets the current configuration settings for the *major* and *minor* device numbers as indicated and displays the **autopush** modules associated with them for the character-special device **/dev/term/a**:

```
example# autopush -g -M 29 -m 0
Major  Minor  Lastminor  Modules
  29    0        1      ldterm ttcompat
```

FILES

/etc/iu.ap

SEE ALSO

bdconfig(1M), **ttymon(1M)**, **ldterm(7M)**, **sad(7D)**, **streamio(7I)** **ttcompat(7M)**
STREAMS Programming Guide

NAME	bdconfig – configures the bd (buttons and dials) stream
SYNOPSIS	bdconfig [startup] [off] [on] [<i>term</i>] [status] [verbose]
AVAILABILITY	SUNWdialh
DESCRIPTION	bdconfig is responsible for configuring the autopush facility and defining to the system what serial device to use for the bd stream.
OPTIONS	<p>If no options are given, then an interactive mode is assumed. In this mode the current status is presented along with this usage line, and a series of interactive questions asked to determine the user's desires.</p> <p>Root privilege is required to change the configuration. The status option does not require root privilege. bdconfig can be installed as a setuid root program.</p> <p>The non-interactive options below can be given in any order.</p> <p><i>term</i> Specify to the system the serial device for bd use. This option implies the on option unless the off option is present.</p> <p>off Reconfigure the configured term for tty use.</p> <p>on Reconfigure the configured term for bd use. If <i>term</i> has not been previously specified, interactive questions are asked to determine the user's desires.</p> <p>startup Configure as was last configured before the system went down. This option is used by the startup script, and precludes the use of the on, off, and <i>term</i> options. This option implies non-interactive mode.</p> <p>status Emit the current configuration in terms of the words used as options: off, on, <i>/dev/term/a</i>, <i>/dev/term/b</i>, and so forth. This option implies non-interactive mode.</p> <p>verbose bdconfig describes what it finds and what it is doing.</p>
EXIT STATUS	bdconfig returns 0 on success, 1 on general error, and 2 on argument error.
SEE ALSO	autopush(1M) , x_buttonstest(6) , x_dialstest(6) , bd(7M) , sad(7D) , streamio(7I)
NOTES	<p>All bdconfig does is configure the AUTOPUSH facility. bdconfig does not actually manipulate the serial port or stream in any way. Only the first open of a dismantled stream will see the effects of a previously run bdconfig.</p> <p>bdconfig is silent except for error messages unless:</p> <ul style="list-style-type: none"> a) invoked with no args: status / usage line emitted b) interactive modes are invoked as described above c) the verbose option is used

BUGS The interface does not support more than one dialbox and one buttonbox, both of which must be on the same serial device.

There should be a library routine to read, parse, and validate records in the `iu.ap` file, so that **bdconfig** could return to the appropriate record in `iu.ap` as the default configuration.

NAME	boot – start the system kernel or a standalone program
SPARC SYNOPSIS	ok boot [<i>OBP names</i>] [<i>filename</i>] [<i>boot-flags</i>] > b [<i>device</i> [<i>(c, u, p)</i>]] [<i>filename</i>] [-a] <i>boot-flags</i>
x86 SYNOPSIS	> b [<i>filename</i>] [<i>boot-args</i>] > i
DESCRIPTION	Bootstrapping is the process of loading and executing a standalone program. For the purpose of this discussion, bootstrapping means the process of loading and executing the bootable operating system. Typically, the standalone program is the operating system kernel (see kernel(1M)), but any standalone program can be booted instead. As an example, on a SPARC system, the diagnostic monitor for a machine is a good example of a standalone program other than the operating system that can be booted.
SPARC Bootstrap Procedure	<p>On SPARC systems, the bootstrap procedure on most machines consists of the following basic phases.</p> <p>After the machine is turned on the system firmware (in PROM) executes power-on self test (POST). The form and scope of these tests depends on the version of the firmware in your system.</p> <p>After the tests have been completed successfully, the firmware attempts to autoboot if the appropriate flag has been set in the non-volatile storage area used by the firmware. The name of the file to load, and the device to load it from can also be manipulated.</p> <p>These flags and names can be set using the eeeprom(1M) command from the shell, or by using PROM commands from the 'ok' prompt after the system has been halted.</p> <p>The second level program is either ufsboot, (when booting from a disk) or inetboot (when booting across the network). When booting from a disk, the PROM assumes that the primary bootblock resides in blocks 1 to 15 of the local disk. When booting over the network, the PROM makes a reverse ARP request and when it receives a reply, the PROM makes a TFTP request to the server that responded and fetches inetboot across the network and executes it. Inetboot also makes another reverse ARP request, then uses the bootparams protocol to locate its root filesystem. It then fetches the kernel across the network using the NFS protocol and then executes it.</p> <p>If the pathname to the standalone is relative (does not begin with a slash), the second level boot will look for the standalone in its platform-dependent location, that is, it prepends /platform/platform-name to the name of the standalone. See filesystem(5). If the pathname is absolute, boot will use the specified path. The boot program then loads the standalone at the appropriate address, and then transfers control.</p> <p>If the standalone is identified as a dynamically-linked executable, boot will load the interpreter (linker/loader) as indicated by the executable format and then transfer control to the interpreter. If the standalone is statically-linked, it will jump directly to the standalone.</p>

Typically, the standalone program is the kernel. Once the program is loaded, it starts the UNIX system, mounts the necessary filesystems (see **vfstab(4)**), and runs **/sbin/init** to bring the system to the "initdefault" state specified in **/etc/inittab** (see **inittab(4)**).

x86 Bootstrap Procedure

On x86 systems, the bootstrapping process consists of two conceptually distinct phases, primary boot and secondary boot. The primary boot is implemented in the BIOS ROM on the system board, in BIOS extensions in ROMs on peripheral boards, and sometimes in an executive program and BIOS extensions on a boot diskette. It is distinguished by its ability to control the installed peripheral devices and to provide I/O services through software interrupts. It begins the booting process by loading the first physical sector from a mass storage device or by loading a program over the LAN. The primary boot is implemented in x86 real-mode code.

The secondary boot is loaded by the primary boot and uses I/O services provided by the primary boot. It is implemented in 32-bit, paged, protected mode code, and it is capable of reading files from a file system. There are two separate versions of the secondary boot:

ufsboot boots from a UFS file system on a hard disk or a CD

inetboot boots over a LAN using the NFS protocol

All versions of the secondary boot contain a built-in shell-like command interpreter. The command language is defined below. The interpreter reads the script in **/etc/bootrc**, which controls the booting process. This file can be modified to change defaults or to adapt to a specific machine.

The standard **/etc/bootrc** script prompts the user to enter a **b** character to boot with specified options, an **i** character to invoke the interpreter interactively, or any other character to boot the default kernel. Once the kernel is loaded, it starts the operating system, loads the necessary modules, mounts the necessary filesystems (see **vfstab(4)**), and runs **/sbin/init** to bring the system to the "initdefault" state specified in **/etc/inittab** (see **inittab(4)**).

SPARC OPTIONS

OBP names Specify the open boot prom designations. For example, on Desktop SPARC systems, the designation **/sbus/esp@0,800000/sd@3,0:a** indicates a SCSI disk (sd) at target 3, lun0 on the SCSI bus, with the esp host adapter plugged into slot 0.

filename Name of a standalone program to boot. The default is to boot **/platform/platform-name/kernel/unix** from the root partition, but you can specify another program, such as **/stand/diag** on the command line. Some versions of the firmware allow the default filename to be saved in the non-volatile storage area of the system.

-a The boot program interprets this flag to mean **ask me**, and so it prompts for the name of the standalone. The **'-a'** flag is then handed onto the standalone.

x86 OPTIONS	<p><i>boot-flags</i> The boot program passes all <i>boot-flags</i> to <i>filename</i>. They are not interpreted by boot. See the kernel(1M) and kadb(1M) manual pages for information on the options available with the default standalone program, kernel/unix.</p> <p><i>filename</i> Name of a standalone program to boot. The default is to boot /platform/platform-name/kernel/unix from the root partition, but you can specify another program on the command line.</p> <p><i>boot-args</i> The boot program passes all <i>boot-args</i> to <i>filename</i>. They are not interpreted by boot. See the kernel(1M) and kadb(1M) manual pages for information on the options available with the kernel.</p>
x86 BOOT SEQUENCE DETAILS	<p>After a PC-compatible machine is turned on, the system firmware in the BIOS ROM executes a power-on self test (POST), runs BIOS extensions in peripheral board ROMs, and invokes software interrupt INT 19h, Bootstrap. The INT 19h handler typically performs the standard PC-compatible boot, which consists of trying to read the first physical sector from the first diskette drive, or, if that fails, from the first hard disk. The processor then jumps to the first byte of the sector image in memory.</p>
x86 Hard Disk Boot	<p>The first sector on a hard disk contains the master boot block, which contains the master boot program and the FDISK table, named for the PC program that maintains it. The master boot finds the active partition in the FDISK table, loads its first sector, and jumps to its first byte in memory. This completes the standard PC-compatible hard disk boot sequence.</p> <p>The x86 FDISK partition begins with a one-cylinder boot slice, which contains the partition boot program (pboot) in the first sector, the standard Solaris disk label and volume table of contents (VTOC) in the second and third sectors, and the bootblk program in the fourth and subsequent sectors. When the Solaris FDISK partition is the active partition, the master boot program (mboot) reads the partition boot program in the first sector into memory and jumps to it. It in turn reads the bootblk program into memory and jumps to it. If the drive contains multiple FDISK partitions, the user is given the opportunity to reboot another partition. bootblk reads ufsboot from the file system in the root slice and jumps to its first byte in memory.</p> <p>Like all versions of the secondary boot, ufsboot switches the processor to 32-bit, paged, protected mode and performs some limited machine initialization. It then invokes its command interpreter, which interprets /etc/bootrc.</p>
x86 CD-ROM Boot	<p>The standard BIOS ROM and BIOS extensions on SCSI Host Bus Adapters (HBAs) do not support booting from CD-ROM. A special Solaris boot diskette is provided for booting from CD-ROM or other devices not supported by the BIOS. It is typically used to install the Solaris operating system from CD-ROM, although it can also be used for system maintenance or network booting.</p> <p>This diskette is in the form of a standard DOS diskette, but is not a DOS system disk. It contains an executive program and BIOS extensions that support booting over the network or reading CD-ROMs through various SCSI HBAs. When the diskette is booted,</p>

the executive program loads and runs each of the BIOS extension files. If an extension detects the presence of the device it supports, it installs itself in memory and chains its service routine to the appropriate software interrupt.

After the extensions have all been loaded, the executive displays a menu of bootable devices. The user chooses the device to be booted, typically the CD.

Once the bootable device has been selected, the boot process proceeds in the same manner as the hard disk boot, i.e., the executive reads the first physical sector from the device and jumps to its first byte.

x86 Network Booting

When booting is to occur over the network, a boot request multicast packet using the Remote Program Load (RPL) protocol is generated by the PC. Somewhere on the same network segment, an RPL server is listening to these boot requests. The information contained in the request is validated and a reply is sent back to the PC. The PC then requests bootfiles to be downloaded to its memory. These bootfiles have been previously configured for this PC. Once downloading has finished, the PC begins executing these programs, which initiate running of the secondary boot program **inetboot**. Like other versions of the secondary boot, **inetboot** switches the processor to 32-bit, paged, protected mode and performs some limited machine initialization. It then issues Reverse Address Resolution Protocol (RARP) requests to find out its own IP address and a **boot-params** RPC to find out which server to mount as its root file system using NFS. When the root has been mounted, it then invokes its command interpreter, which interprets **/etc/bootrc**.

Secondary Boot Programming Language for x86

The wide range of hardware that must be supported on x86 demands great flexibility in the booting process. This flexibility is achieved in part by making the secondary boot programmable. The secondary boot contains an interpreter that accepts a simple command language similar to those of **sh** and **csh**. The primary differences are that pipelines, loops, standard output, and output redirection are not supported.

x86 Lexical Structure

The boot interpreter splits input lines into words separated by blanks and tabs. The metacharacters are dollar sign (\$), single-quote ('), double-quote ("), number sign (#), new-line, and backslash (\). The special meaning of metacharacters can be avoided by preceding them with a backslash. A new-line preceded by a backslash is treated as a blank. A number sign introduces a comment, which continues to the next new-line.

A string enclosed in a pair of single-quote or double-quote characters forms all or part of a single word. White space and new-line characters within a quoted string become part of the word. Characters within a quoted string can be quoted by preceding them with a backslash character; thus a single-quote character can appear in a single-quoted string by preceding it with a backslash. Two backslashes produce a single backslash, and a new-line preceded by a backslash produces a new-line in the string.

x86 Variables

The boot maintains a set of variables, each of which has a string value. The first character of a variable name must be a letter, and subsequent characters can be letters, digits, or underscores. The **set** command creates a variable and/or assigns a value to it, or displays the values of variables. The **unset** command deletes a variable.

Variable substitution is performed when the interpreter encounters a dollar-sign that is not preceded by a backslash. The variable name following the dollar sign is replaced by the value of the variable, and parsing continues at the beginning of the value. Variable substitution is performed in double-quoted strings, but not in single-quoted strings. A variable name can be enclosed in braces to separate it from following characters.	
x86 Commands	A command is a sequence of words terminated by a new-line character. The first word is the name of the command and subsequent words are arguments to the command. All commands are built-in commands. Standalone programs are executed with the run command.
x86 Conditional Execution of Commands	<p>Commands can be conditionally executed by surrounding them with the if, elseif, else, and endif commands:</p> <pre style="margin-left: 2em;"> if <expr1> ... elseif <expr2> ... elseif <expr3> ... else ... endif </pre> <p>An if block may be embedded in other if blocks.</p>
x86 Expressions	<p>The set, if, and elseif commands evaluate arithmetic expressions with the syntax and semantics of the C programming language. The <code> </code>, <code>&&</code>, <code> </code>, <code>^</code>, <code>&</code>, <code>==</code>, <code>!=</code>, <code><</code>, <code>></code>, <code><=</code>, <code>>=</code>, <code>>></code>, <code><<</code>, <code>+</code>, <code>-</code>, <code>*</code>, <code>/</code>, <code>%</code>, <code>~</code>, and <code>!</code> operators are accepted, as are <code>(</code>, <code>)</code>, and comma. Signed 32-bit integer arithmetic is performed.</p> <p>Expressions are parsed after the full command line has been formed. Each token in an expression must be a separate argument word, so blanks must separate all tokens on the command line.</p> <p>Before an arithmetic operation is performed on an operand word, it is converted from a string to a signed 32-bit integer value. After an optional leading sign, a leading 0 produces octal conversion and a leading 0x or 0X produces hexadecimal conversion. Otherwise, decimal conversion is performed. A string that is not a legal integer is converted to zero.</p> <p>Several built-in functions for string manipulation are provided. Built-in function names begin with a dot. String arguments to these functions are not converted to integers. To cause an operator, e.g., <code>-</code>, to be treated as a string, it must be preceded by a backslash, and that backslash must be quoted with another backslash. Also be aware that a null string can produce a blank argument, and thus an expression syntax error. For example:</p> <pre style="margin-left: 2em;"> if .strneq (\${usrarg}X , \- , 1) </pre>

	is the safe way to test whether the variable <i>usrarg</i> starts with a <code>-</code> , even if it could be null.
x86 I/O	<p>The boot interpreter takes its input from the system console or from one or more files. The source command causes the interpreter to read a file into memory and begin parsing it. The console command causes the interpreter to take its input from the system console. Reaching EOF causes the interpreter to resume parsing the previous input source. CTRL-D entered at the beginning of console line is treated as EOF.</p> <p>The echo command writes its arguments to the display. The read command reads the system console and assigns word values to its argument variables.</p>
x86 Debugging	<p>The verbose command turns verbose mode on and off. In verbose mode, the interpreter displays lines from the current source file and displays the command as actually executed after variable substitution.</p> <p>The singlestep command turns singlestep mode on and off. In singlestep mode, the interpreter displays step ? before processing the next command, and waits for keyboard input, which is discarded. Processing proceeds when ENTER is pressed. This allows slow execution in verbose mode.</p>
x86 Initialization	When the interpreter is first invoked by the boot, it begins execution of a compiled-in initialization string. This string typically consists of "source /etc/bootrc\n" to run the boot script in the root file system.
x86 Communication With Standalone Programs	The boot passes information to standalone programs through arguments to the run command. A standalone program can pass information back to the boot by setting a boot interpreter variable using the var_ops() boot service function. It can also pass information to the kernel using the setprop() boot service function. The whoami property is set to the name of the standalone program.
x86 Built-in Commands	<p>console Interpret input from the console until CTRL-D.</p> <p>echo arg1 ... Display the arguments separated by blanks and terminate with a new-line.</p> <p>echo -n arg1 ... Display the arguments separated by blanks, but do not terminate with a new-line.</p> <p>else elseif endif See if.</p> <p>getprop <i>propname varname</i> Assign the value of property <i>propname</i> to the variable <i>varname</i>. A property value of length zero produces a null string. If the property does not exist, the variable is not set.</p>

getproplen *propname varname*

Assign the length in hexadecimal of the value of property *propname* to the variable *varname*. Property value lengths include the terminating null. If the property does not exist, the variable is set to 0xFFFFFFFF (-1).

if *<expr>*

If the expression *expr* is true, execute instructions to the next **elseif**, **else**, or **endif**. If **expr** is false, do not execute the instructions.

elseif *<expr>*

If the preceding **if** and **elseif** commands all failed, and *expr* is true, execute instructions to the next **elseif**, **else**, or **endif**. Otherwise, do not execute the instructions.

else If the preceding **if** and **elseif** commands all failed, execute instructions to the next **elseif**, **else**, or **endif**. Otherwise, do not execute the instructions.

endif Revert to the execution mode of the surrounding block.

help Display a help screen that contains summaries of all available boot shell commands.

read *name1 . . .*

Read a line from the console, break it into words, and assign them as values to the variables *name1*, etc.

readt *time . . .*

Same as **read**, but timeout after *time* seconds.

run *name arg1 . . .*

Load and transfer control to the standalone program *name*, passing it *arg1* and further arguments.

set Display all the current variables and their values.

set *name*

Set the value of the variable *name* to the null string.

set *name word*

Set the value of the variable *name* to *word*.

set *name <expr>*

Set the value of the variable *name* to the value of *expr*. *expr* must consist of more than one word. The value is encoded in unsigned hexadecimal, so that -1 is represented by 0xFFFFFFFF.

setcolor

Set the text mode display attributes. Allowable colors are *black*, *blue*, *green*, *cyan*, *red*, *magenta*, *brown*, *white*, *gray*, *lt_blue*, *lt_green*, *lt_cyan*, *lt_red*, *lt_magenta*, *yellow*, and *hi_white*.

setprop *propname word*

Set the value of the property *propname* to *word*.

singlestep**singlestep on**

Turn on singlestep mode, in which the interpreter displays **step ?** before each command is processed, and waits for keyboard input. Press ENTER to execute the next command.

singlestep off

Turn off singlestep mode.

source *name*

Read the file *name* into memory and begin to interpret it. At EOF, return to the previous source of input.

unset *name*

Delete the variable name.

verbose**verbose on**

Turn on verbose mode, which displays lines from source files and commands to be executed.

verbose off

Turn off verbose mode.

**x86 Built-in
Functions**

The following built-in functions are accepted within expressions:

.strcmp(*string1*, *string2*)

Returns an integer value that is less than, equal to, or greater than zero, as *string1* is lexicographically less than, equal to, or greater than *string2*.

.strncmp(*string1*, *string2*, *n*)

Returns an integer value that is less than, equal to, or greater than zero, as *string1* is lexicographically less than, equal to, or greater than *string2*. At most, *n* characters are compared.

.streq (*string1*, *string2*)

Returns true if *string1* is equal to *string2*, and false otherwise.

.strneq (*string1*, *string2*, *n*)

Returns true if *string1* is equal to *string2*, and false otherwise. At most, *n* characters are compared.

.strfind (*string*, *addr*, *n*)

Scans *n* locations in memory starting at *addr*, looking for the beginning of *string*. The *string* in memory need not be null-terminated. Returns true if *string* is found, and false otherwise. **.strfind** can be used to search for strings in the ROM BIOS and BIOS extensions that identify different machines and peripheral boards.

FILES**/platform/*platform-name*/ufsboot**

second level program to boot from a disk or CD.

/platform/*platform-name*/kernel/unix

default program to boot system.

/etc/inittab

table in which the "initdefault" state is specified.

/sbin/init program that brings the system to the "initdefault" state.

x86 Only **/etc/bootrc** script that controls the booting process.

SEE ALSO **uname(1)**, **eeprom(1M)**, **init(1M)**, **installboot(1M)**, **kadb(1M)**, **kernel(1M)**, **shutdown(1M)**, **inittab(4)**, **vfstab(4)**

System Administration Guide, Volume I

x86 Only **rpld(1M)**

WARNINGS The boot program is not smart enough to know which files can be used as bootable programs. If the booting of a file that is not bootable is requested, the boot program loads it and branches to it. What happens after that is unpredictable.

NOTES Because the “-” key on national language keyboards has been moved, an alternate key must be used to supply arguments to the **boot** command on an x86 system using these keyboards. Use the “-” on the numeric keypad. The specific language keyboard and the alternate key to be used in place of the “-” during bootup is shown below.

Keyboard	Substitute Key
Italy	^
Spain	^
Sweden	+
France	?
Germany	?

For example, **b -r** would be typed as **b +r** on Swedish keyboards, although the screen display will show as **b -r**.

NAME	bsmconv, bsmunconv – enable/disable the Basic Security Module (BSM) on Solaris
SYNOPSIS	<code>/etc/security/bsmconv [rootdir ...]</code> <code>/etc/security/bsmunconv [rootdir ...]</code>
DESCRIPTION	<p>The bsmconv and bsmunconv scripts are used to enable or disable the BSM features on a Solaris system. The optional argument rootdir is a list of one or more root directories of diskless clients which have already been configured via the Host Manager, see admintool(1M).</p> <p>To enable or disable BSM on a diskless client, a server, or a stand-alone system, logon as super-user to the system being converted and use the bsmconv or bsmunconv commands without any options.</p> <p>To enable or disable BSM on a diskless client from that client's server, logon to the server as super-user and use bsmconv, specifying the root directory of each diskless client you wish to affect. For example, the command</p> <pre>myhost# bsmconv /export/root/client1 /export/root/client2</pre> <p>will enable BSM on the two machines named <i>client1</i> and <i>client2</i>. While the command</p> <pre>myhost# bsmconv</pre> <p>will enable BSM only on the machine called myhost. It is no longer necessary to enable BSM on both the server and its diskless clients.</p> <p>After running bsmconv the system can be configured by editing the files in /etc/security. Each diskless client has its own copy of configuration files in its root directory. You may wish to edit these files before rebooting each client.</p> <p>Following the completion of either script, the affected system(s) should be rebooted to allow the auditing subsystem to come up properly initialized.</p>
SEE ALSO	admintool (1M) auditd (1M), audit_startup (1M), audit.log (4), audit_control (4)

NAME	cachefslog – Cache File System logging
SYNOPSIS	cachefslog [-f logfile -h] <i>path</i>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>The cachefslog command displays where cacheFS statistics are being logged, and optionally sets where cacheFS statistics are being logged, or halts logging for a cache specified by <i>path</i>. <i>path</i> may be a cache or a mountpoint of a cache filesystem. All filesystems cached under the same cache as <i>path</i> will be logged.</p> <p>You must be super-user to use the -f and -h options.</p>
OPTIONS	<p>The following options are supported:</p> <p>-f logfile Specify the log file to be used.</p> <p>-h Halt logging.</p>
EXAMPLES	<p>The example below checks if the directory /home/sam is being logged:</p> <pre>example% cachefslog /home/sam not logged: /home/sam</pre> <p>The example below changes the <i>logfile</i> of /home/sam to /var/tmp/samlog:</p> <pre>example# cachefslog -f /var/tmp/samlog /home/sam /var/tmp/samlog: /home/sam</pre> <p>The example below verifies the change of the previous example:</p> <pre>example% cachefslog /home/sam /var/tmp/samlog: /home/sam</pre> <p>The example below halts logging for the /home/sam directory:</p> <pre>example# cachefslog -h /home/sam not logged: /home/sam</pre>
EXIT STATUS	<p>The following exit values are returned:</p> <p>0 success</p> <p>non-zero an error has occurred.</p>
SEE ALSO	cachefswssize(1M) , cachefsstat(1M)
DIAGNOSTICS	<p>Invalid path</p> <p>It is illegal to specify a path within a cache filesystem.</p>

NAME	cachefsstat – Cache File System statistics
SYNOPSIS	cachefsstat [<i>-z</i>] [<i>path...</i>]
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>The cachefsstat command displays statistical information about the cache filesystem mounted on <i>path</i>. The statistical information includes cache hits and misses, consistency checking, and modification operations. If <i>path</i> is not specified, all mounted cache filesystems are used.</p> <p>cachefsstat can also be used to reinitialize this information (see <i>-z</i> option). The statistical information has the following format:</p> <pre> <cache hit rate> <consistency checks> <modifies> where: <i>hit rate</i> The percentage of cache hits over the total number of attempts, followed by the actual numbers of hits and misses. <i>consistency checks</i> The number of consistency checks performed, followed by the number that passed, and the number that failed. <i>modifies</i> The number of modify operations, including writes, creates, etc.</pre>
OPTIONS	<p>The following option is supported:</p> <p><i>-z</i> Zero (reinitialize) statistics. Execute cachefsstat -z before executing cachefsstat again to gather statistics on the cache performance. This option can only be use by the super-user. The statistics printed reflect those just before the statistics are re-initialized.</p>
EXAMPLES	<pre> example% cachefsstat /home/sam cache hit rate: 73% (1234 hits, 450 misses) consistency checks: 700 (650 pass, 50 fail) modifies: 321</pre>
EXIT STATUS	<p>The following exit values are returned:</p> <p>0 success</p> <p>non-zero an error has occurred.</p>
SEE ALSO	cachefslog (1M)

NAME	cachefswsize – Determine working set size for cachefs
SYNOPSIS	cachefswsize <i>logfile</i>
AVAILABILITY	SUNWcsu
DESCRIPTION	The cachefswsize command displays the workspace size determined from <i>logfile</i> . This includes the amount of cache space needed for each filesystem that was mounted under the cache, as well as a total.
EXAMPLES	<pre> example% cachefswsize /var/tmp/samlog /home/sam end size: 10688k high water size: 10704k /foo end size: 128k high water size: 128k /usr/dist end size: 1472k high water size: 1472k total for cache initial size: 110960k end size: 12288k high water size: 12304k </pre>
EXIT STATUS	<p>The following exit values are returned:</p> <p>0 success</p> <p>non-zero an error has occurred.</p>
SEE ALSO	cachefslog(1M)
DIAGNOSTICS	<p>problems were encountered writing log file There were problems encountered when the kernel was writing the logfile. The most common problem is running out of disk space.</p> <p>invalid log file The logfile isn't a valid logfile or was created with a newer version of Solaris than the one where cachefswsize is running.</p>

NAME	captainfo – convert a termcap description into a terminfo description
SYNOPSIS	captainfo [-1] [-v . . .] [-V] [-w <i>width</i>] <i>filename</i> . . .
DESCRIPTION	<p>captainfo looks in <i>filename</i> for termcap descriptions. For each one found, an equivalent terminfo description is written to standard output, along with any comments found. A description which is expressed as relative to another description (as specified in the termcap tc = field) is reduced to the minimum superset before being displayed.</p> <p>If no <i>filename</i> is given, then the environment variable TERMCAP is used for the filename or entry. If TERMCAP is a full pathname to a file, only the terminal whose name is specified in the environment variable TERM is extracted from that file. If the environment variable TERMCAP is not set, then the file /usr/share/lib/termcap is read.</p>
OPTIONS	<p>-1 Display the fields one to a line. Otherwise, the fields are printed several to a line, with a maximum width of 60 characters.</p> <p>-v Display tracing information on the standard error as the program runs. Specifying additional -v options displays more detailed information.</p> <p>-V Display the version of the program in use on the standard error and then exit.</p> <p>-w <i>width</i> Change the output to <i>width</i> characters.</p>
FILES	<p>/usr/share/lib/terminfo/?/* compiled terminal description database</p> <p>/usr/share/lib/termcap</p>
SEE ALSO	infocmp(1M) , curses(3X) , terminfo(4)
NOTES	captainfo should be used to convert termcap entries to terminfo entries because the termcap database may not be supplied in future releases.

NAME	catman – create the cat files for the manual														
SYNOPSIS	<code>/usr/bin/catman [-n] [-p] [-t] [-w] [-M <i>directory</i>] [-T <i>macro-package</i>] [<i>sections</i>]</code>														
DESCRIPTION	<p>catman creates the preformatted versions of the on-line manual from the nroff(1) input files. Each manual page is examined and those whose preformatted versions are missing or out of date are recreated. If any changes are made, catman recreates the windex database.</p> <p>If there is one parameter not starting with a ‘-’, it is taken to be a space separated list of manual sections to create. For example</p> <p style="text-align: center;">catman 1 2 3</p> <p>only updates manual sections 1, 2, and 3.</p> <p>If an unformatted source file contains only a line of the form .so manx/yyy.x, a symbolic link is made in the catx or fmtx directory to the appropriate preformatted manual page. This feature allows easy distribution of the preformatted manual pages among a group of associated machines (for example, with rdist(1)), since it makes the directories of preformatted manual pages self-contained and independent of the unformatted entries.</p>														
OPTIONS	<p>-n Do not create (or recreate) the windex database.</p> <p>-p Print what would be done instead of doing it.</p> <p>-t Create troffed entries in the appropriate fmt subdirectories instead of nroffing into the cat subdirectories.</p> <p>-w Only create the windex database that is used by whatis(1) and the man(1) -f and -k options. No manual reformatting is done.</p> <p>-M <i>directory</i> Update manual pages located in the specified directory (/usr/share/man by default).</p> <p>-T <i>macro-package</i> Use <i>macro-package</i> in place of the standard manual page macros.</p>														
ENVIRONMENT	<p>TROFF The name of the formatter to use when the -t flag is given. If not set, troff(1) is used.</p>														
FILES	<table border="0"> <tr> <td>/usr/share/man</td> <td>default manual directory location</td> </tr> <tr> <td>/usr/share/man/man?/*.*</td> <td>raw (nroff input) manual sections</td> </tr> <tr> <td>/usr/share/man/cat?/*.*</td> <td>preformatted nroffed manual pages</td> </tr> <tr> <td>/usr/share/man/fmt?/*.*</td> <td>preformatted troffed manual pages</td> </tr> <tr> <td>/usr/share/man/windex</td> <td>table of contents and keyword database</td> </tr> <tr> <td>/usr/lib/makewhatis</td> <td>command script to make windex database</td> </tr> <tr> <td>/usr/share/lib/tmac/an</td> <td>default macro package</td> </tr> </table>	/usr/share/man	default manual directory location	/usr/share/man/man?/*.*	raw (nroff input) manual sections	/usr/share/man/cat?/*.*	preformatted nroffed manual pages	/usr/share/man/fmt?/*.*	preformatted troffed manual pages	/usr/share/man/windex	table of contents and keyword database	/usr/lib/makewhatis	command script to make windex database	/usr/share/lib/tmac/an	default macro package
/usr/share/man	default manual directory location														
/usr/share/man/man?/*.*	raw (nroff input) manual sections														
/usr/share/man/cat?/*.*	preformatted nroffed manual pages														
/usr/share/man/fmt?/*.*	preformatted troffed manual pages														
/usr/share/man/windex	table of contents and keyword database														
/usr/lib/makewhatis	command script to make windex database														
/usr/share/lib/tmac/an	default macro package														

SEE ALSO	apropos(1), man(1), nroff(1), rdist(1), rm(1), troff(1), whatis(1)
DIAGNOSTICS	man?/xxx.? (.so'ed from man?/yyy.?): No such file or directory The file outside the parentheses is missing, and is referred to by the file inside them. target of .so in man?/xxx.? must be relative to /usr/man catman only allows references to filenames that are relative to the directory /usr/man . opendir:man?: No such file or directory A harmless warning message indicating that one of the directories catman normally looks for is missing. *.*: No such file or directory A harmless warning message indicating catman came across an empty directory.
WARNINGS	If a user, who has previously run catman to install the cat? directories, upgrades the operating system, the entire cat? directory structure should be removed prior to running catman . See rm(1) .
NOTES	If the -n option is specified, the /usr/share/man/windex database is not created and the apropos , whatis , man -f , and man -k commands will fail. If the -M option is specified, the directory argument must not contain a ' , since ' is used to delineate section numbers. See man(1) .

NAME	cfsadmin – administer disk space used for caching file systems with the Cache File-System (CacheFS)
SYNOPSIS	<pre> cfsadmin -c [-o <i>cacheFS-parameters</i>] <i>cache_directory</i> cfsadmin -d [<i>cache_ID</i> all] <i>cache_directory</i> cfsadmin -l <i>cache_directory</i> cfsadmin -s [<i>mntpt1...</i> all] <i>cache_directory</i> cfsadmin -u [-o <i>cacheFS-parameters</i>] </pre>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>The cfsadmin command provides the following functions:</p> <ul style="list-style-type: none"> • cache creation • deletion of cached file systems • listing of cache contents and statistics • resource parameter adjustment when the file system is unmounted. <p>For each form of the command, with the except of the -s form, you must specify a cache directory, that is, the directory under which the cache is actually stored. A path name in the front file system identifies the cache directory. For the -s form of the command, you must specify a mount point.</p> <p>You can specify a cache ID when you mount a file system with CacheFS, or you can let the system generate one for you. The -l option includes the cache ID in its listing of information. You must know the cache ID to delete a cached file system.</p>
OPTIONS	<p>-c Create a cache under the directory specified by <i>cache_directory</i>. This directory must not exist prior to cache creation.</p> <p>-d Remove the file system whose cache ID you specify and release its resources, or remove all file systems in the cache by specifying all. After deleting a file system from the cache, you must run the fsck_cachefs(1M) command to correct the resource counts for the cache.</p> <p>-l List file systems stored in the specified cache, as well as statistics about them. Each cached file system is listed by cache ID. The statistics document resource utilization and cache resource parameters.</p> <p>-s Request a consistency check on the specified file system (or all cachefs mounted file systems). The -s option will only work if the cache file system was mounted with demandconst enabled (see mount_cachefs(1M)). Each file in the specified cache file system is checked for consistency with its corresponding file in the back file system. Note that the consistency check is performed file by file as files are accessed. If no files are accessed, no checks are performed. Use of this option will not result in a sudden "storm" of consistency checks.</p> <p>-u Update resource parameters of the specified cache directory. Parameter values can only be increased. To decrease the values, you must remove</p>

the cache and recreate it. All file systems in the cache directory must be unmounted when you use this option. Changes will take effect the next time you mount any file system in the specified cache directory. The **-u** option with no **-o** option sets all parameters to their default values.

CacheFS Resource Parameters

You can specify the following cacheFS resource parameters as arguments to the **-o** option. Separate multiple parameters with commas.

- maxblocks=*n*** Maximum amount of storage space that CacheFS can use, expressed as a percentage of the total number of blocks in the front file system. If CacheFS does not have exclusive use of the front file system, there is no guarantee that all the space the **maxblocks** parameter allows will be available. The default is **90**.
- minblocks=*n*** Minimum amount of storage space, expressed as a percentage of the total number of blocks in the front file system, that CacheFS is always allowed to use without limitation by its internal control mechanisms. If CacheFS does not have exclusive use of the front file system, there is no guarantee that all the space the **minblocks** parameter attempts to reserve will be available. The default is **0**.
- threshblocks=*n*** A percentage of the total blocks in the front file system beyond which CacheFS cannot claim resources once its block usage has reached the level specified by **minblocks**. The default is **85**.
- maxfiles=*n*** Maximum number of files that CacheFS can use, expressed as a percentage of the total number of inodes in the front file system. If CacheFS does not have exclusive use of the front file system, there is no guarantee that all the inodes the **maxfiles** parameter allows will be available. The default is **90**.
- minfiles=*n*** Minimum number of files, expressed as a percentage of the total number of inodes in the front file system, that CacheFS is always allowed to use without limitation by its internal control mechanisms. If CacheFS does not have exclusive use of the front file system, there is no guarantee that all the inodes the **minfiles** parameter attempts to reserve will be available. The default is **0**.
- threshfiles=*n*** A percentage of the total inodes in the front file system beyond which CacheFS cannot claim inodes once its usage has reached the level specified by **minfiles**. The default is **85**.
- maxfilesize=*n*** Largest file size, expressed in megabytes, that CacheFS is allowed to cache. The default is **3**.

Note: You cannot decrease the block or inode allotment for a cache. To decrease the size of a cache, you must remove it and create it again with different parameters.

EXAMPLES

The following example creates a cache directory named **/cache**:

```
example# cfsadmin -c /cache
```

The following example creates a cache named **/cache1** that can claim a maximum of 60 percent of the blocks in the front file system, can use 40 percent of the front file system blocks without interference by CacheFS internal control mechanisms, and has a threshold value of 50 percent. The threshold value indicates that after CacheFS reaches its guaranteed minimum, it cannot claim more space if 50 percent of the blocks in the front file system are already used.

```
example# cfsadmin -c -o maxblocks=60,minblocks=40,  
thresblocks=50 /cache1
```

The following example changes the **maxfilesize** parameter for the cache directory **/cache2** to 2 megabytes:

```
example# cfsadmin -u -o maxfilesize=2 /cache2
```

The following example lists the contents of a cache directory named **/cache3** and provides statistics about resource utilization:

```
example# cfsadmin -l /cache3
```

The following example removes the cached file system with cache ID **23** from the cache directory **/cache3** and frees its resources (the cache ID is part of the information returned by **cfsadmin -l**):

```
example# cfsadmin -d 23 /cache3
```

The following example removes all cached file systems from the cache directory **/cache3**:

```
example# cfsadmin -d all /cache3
```

The following example checks for consistency all filesystems mounted with **demandconst** enabled. No errors will be reported if no **demandconst** filesystems were found.

```
example# cfsadmin -s all
```

SEE ALSO

fsck_cachefs(1M), **mount_cachefs(1M)**

NAME	cg14config – configure the cgfourteen device
SYNOPSIS	<code>/usr/platform/platform-name/sbin/cg14config [-d device] [-r resolution] [-g gammavalue] [-G gammafile] [-u degammavalue] [-U degammafile]</code>
AVAILABILITY	SUNWkvm
DESCRIPTION	<p>cg14config sets up state on the selected cgfourteen device. <i>platform-name</i> can be found using the <code>-i</code> option of uname(1).</p> <p>cg14config is supported only on Desktop SPARCsystems with SX graphics option. The interface, output, and command location are uncommitted and subject to change in future releases.</p>
OPTIONS	<p><code>-d device</code> Use <i>device</i> as the cgfourteen device to configure. Default is <code>/dev/fb</code>.</p> <p><code>-r resolution</code> Use <i>resolution</i> as the desired screen resolution. Resolution is specified in terms of screen width and height (in pixels), and vertical refresh (in hz). Available resolutions are:</p> <ul style="list-style-type: none"> 1024x768@60 1024x768@66 1024x768@70 1152x900@66 1152x900@76 1280x1024@66 1280x1024@76 1600x1280@66 1920x1080@72 <p>The default is the value read from the monitor sense codes. Note that some or all of the resolutions above may not be supported by any given monitor. If a programmed resolution is outside of the range of allowable values for a monitor, unpredictable results can occur, including damage to the monitor. Thus, care should be taken when programming the resolution. See <i>Openboot Command Reference</i> for a description of how to reset the console device to the default value if it becomes unusable from programming an unsupported resolution.</p> <p>The <code>-r</code> option is not available when the window system is running.</p> <p><code>-g gammavalue</code> Each entry of the gamma lookup table will be loaded with $\text{entry}^{(1/\text{gammavalue})}$. The gamma lookup table has 256 entries. Default <i>gammavalue</i> is 2.2.</p> <p><code>-G filename</code> Initialize the gamma lookup table with the contents of <i>filename</i>. The format of <i>filename</i> is 256 triplets (red green blue) of non-negative integers separated by NEWLINE characters. The integers must be in the range 0 to 1023, inclusive.</p>

-u degammavalue

Each entry of the degamma lookup table will be loaded with entry *degammavalue*. The degamma lookup table has 256 entries. Default *degammavalue* is **2.2**.

-U filename

Initialize the degamma lookup table with the contents of *filename*. The format of *filename* is 256 entries of non-negative integers separated by NEWLINE characters. The integers must be in the range 0 to 255, inclusive.

EXIT STATUS

cg14config returns **0** on success and a positive integer on failure.

- 1** Selected device is not a cgfourteen device.
- 2** Requested action failed.
- 3** Unsupported resolution.
- 4** Gamma or degamma value out of range.

FILES

/platform/platform-name/kernel/drv/cgfourteen
cgfourteen device driver

SEE ALSO

uname(1), **init(1M)**, **mmap(2)**

Platform Notes: SPARCstation 10SX System Configuration Guide

Openboot Command Reference

NAME	chown – change owner
SYNOPSIS	<code>/usr/ucb/chown [-f] [-R] owner [group] filename ...</code>
DESCRIPTION	<p>chown changes the owner of the <i>filenames</i> to <i>owner</i>. The owner may be either a decimal user ID (UID) or a login name found in the password file. An optional <i>group</i> may also be specified. The group may be either a decimal group ID (GID) or a group name found in the GID file.</p> <p>Only the super-user of the machine where the file is physically located can change owner, in order to simplify accounting procedures.</p>
OPTIONS	<p>-f Do not report errors.</p> <p>-R Recursively descend into directories setting the ownership of all files in each directory encountered. When symbolic links are encountered, their ownership is changed, but they are not traversed.</p>
FILES	<code>/etc/passwd</code> password file
SEE ALSO	<code>chgrp(1)</code> , <code>chown(2)</code> , <code>group(4)</code> , <code>passwd(4)</code>

NAME	chroot – change root directory for a command
SYNOPSIS	<i>/usr/sbin/chroot newroot command</i>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>chroot causes <i>command</i> to be executed relative to <i>newroot</i>. The meaning of any initial slashes (/) in the path names is changed to <i>newroot</i> for <i>command</i> and any of its child processes. Upon execution, the initial working directory is <i>newroot</i>.</p> <p>Notice that redirecting the output of <i>command</i> to a file:</p> <pre>chroot newroot command >x</pre> <p>will create the file <i>x</i> relative to the original root of <i>command</i>, not the new one.</p> <p>The new root path name is always relative to the current root: even if a chroot is currently in effect, the <i>newroot</i> argument is relative to the current root of the running process.</p> <p>This command can be run only by the super-user.</p>
RETURN VALUES	The exit status of chroot is the return value of <i>command</i> .
EXAMPLE	<p>chroot provides an easy way to extract tar files written with absolute filenames to a different location.</p> <pre>example# cp /usr/sbin/static/tar /tmp example# dd if=/dev/nrst0 chroot /tmp tar xvf -</pre> <p>Note that tar is statically linked, so you do not have to copy any shared libraries to the <i>newroot</i> filesystem.</p>
SEE ALSO	cd(1), chroot(2)
NOTES	Exercise extreme caution when referencing device files in the new root file system.

NAME	chrtbl, wchrtbl – generate character classification and conversion tables
SYNOPSIS	chrtbl [<i>filename</i>] wchrtbl [<i>filename</i>]
DESCRIPTION	<p>chrtbl creates character type and numeric layout files for single byte locales. wchrtbl does the same for multibyte locales. The two commands are links to each other.</p> <p>Character classification tables contain information on character attributes, upper- to lowercase conversion, and codeset character width. The LC_CTYPE file is an array of bytes encoded so simple table lookups can determine character type or perform case mapping, using ctype(3C) or wctype (see iswalpha(3I)) library routines. Other routines can find the byte count and screen width of characters in supplementary code sets. The LC_NUMERIC file contains format information for numbers. The first byte specifies the decimal delimiter, and the second byte specifies the thousands separator.</p> <p>Both commands read character classification and conversion information from <i>filename</i> and create three output files in the current directory. If no input file is given, these commands read from standard input. The example section below contains the source <i>filename</i> for ISO 8859-1. For multibyte locales this example needs to be extended.</p>
First Output File	<p>The first output file, [w]ctype.c is a C language source file, which application programs can use as needed. It contains a $(257*2)+7$ byte array generated from processing <i>filename</i>. Review the contents of the C source to verify that the array is set up as planned. The first 257 bytes of the array are used for character classification. Symbols used for initializing these bytes represent character classifications defined in <ctype.h>; for example, _L means a character is lower case and _S _B means the character is both a spacing character and a blank. The second 257 bytes of the array are used for character conversion. These bytes are initialized so that characters without conversion information are converted to themselves. If you provide conversion information, the first value of the pair is stored where the second one would normally be stored, and vice versa. For example, if you provide <0x41 0x61>, then 0x61 is stored where 0x41 would normally be stored, and 0x41 is stored where 0x61 would normally be stored. The last 7 bytes are used for character width information for up to three supplementary code sets.</p>
Second Output File	<p>The second output file is binary data containing the same information, but structured for efficient use by the ctype(3C) and wctype (see iswalpha(3I)) routines. The name of this output file is the value you assign to the keyword LC_CTYPE in <i>filename</i>. The superuser should install this file as /usr/lib/locale/locale/LC_CTYPE/ctype. It must be readable by user, group, and other; execute permission is not necessary. Application programs consult this file when the LC_CTYPE environment is set appropriately, upon calling setlocale(3C).</p>
Third Output File	<p>The third output file is binary data created only if numeric formatting information is specified. The name of this output file is the value you assign to the keyword LC_NUMERIC in <i>filename</i>. The superuser should install this file as</p>

`/usr/lib/locale/locale/LC_NUMERIC`. It must be readable by user, group, and other; execute permission is not necessary. Application programs consult this file when the `LC_NUMERIC` environment is set appropriately, upon calling `setlocale(3C)`.

For supplementary codesets, there are three sets of tables. The first set contains three pointer arrays that point to supplementary codeset information tables. If supplementary codeset information is not specified, the contents of the pointers are zeros. The arrays are full of null pointers. The second set contains three supplementary codeset information tables, each specifying minimum and maximum code values to be classified and converted, and also pointers to character classification and conversion tables. If there is no corresponding table, the contents of the pointers are zeros. The third set contains character classification and conversion tables that contain the same information as the single byte table, except codes are represented as process codes and table size is variable. The characters used for initializing values of the character classification table represent character classifications defined in `<wctype.h>`; `_E1` through `_E8` are for international use and `_E9` through `_E24` are for language-dependent use.

filename Syntax

The syntax of *filename* provides for data file naming, assignment of characters to character classifications, upper- to lower-case mapping, byte and screen widths for up to three supplementary code sets, plus numeric formatting information. The keywords recognized by `[w]chrtbl` are:

<code>LC_CTYPE</code>	name of the first data file created by <code>[w]chrtbl</code>
<code>isupper</code>	character codes classified as upper-case letters
<code>islower</code>	character codes classified as lower-case letters
<code>isdigit</code>	character codes classified as numeric
<code>isspace</code>	character codes classified as spacing (delimiter) characters
<code>ispunct</code>	character codes classified as punctuation characters
<code>isctrl</code>	character codes classified as control characters
<code>isblank</code>	character code for the blank (space) character
<code>isxdigit</code>	character codes classified as hexadecimal digits
<code>ul</code>	relationship between upper- and lower-case characters
<code>cswidth</code>	byte count and screen width information
<code>LC_NUMERIC</code>	name of the second data file created by <code>[w]chrtbl</code>
<code>decimal_point</code>	decimal delimiter, may be <code>\NNN</code> octal or <code>\xNN</code> hexadecimal
<code>thousands_sep</code>	thousands separator, may be <code>\NNN</code> octal or <code>\xNN</code> hexadecimal
<code>LC_CTYPE1</code>	begin definition of supplementary codeset 1
<code>LC_CTYPE2</code>	begin definition of supplementary codeset 2
<code>LC_CTYPE3</code>	begin definition of supplementary codeset 3

isphonogram(iswchar1)	character codes classified as phonograms in supplementary code sets
isideogram(iswchar2)	character codes classified as ideograms in supplementary code sets
isenglish(iswchar3)	character codes classified as English letters in supplementary code sets
isnumber(iswchar4)	character codes classified as numeric in supplementary code sets
isspecial(iswchar5)	character codes classified as special letters in supplementary code sets
iswchar6	character codes classified as other printable letters in supplementary code sets
iswchar7 - iswchar8	reserved for international use
iswchar9 - iswchar24	character codes classified as language-dependent letters/characters

Any lines with a sharp (#) in the first column are treated as comments and are ignored, as are blank lines.

To indicate character codes, use either hexadecimal or octal constants. For example, the letter **a** can be represented as **0x61** in hexadecimal or **0141** in octal. Constants may be separated by one or more spaces and/or tabs. Use a dash (-) to indicate a range of consecutive numbers. Zero or more spaces may separate the dash from its numbers. Use a backslash (\) for line continuation; only the newline is permitted after a backslash. Character codes are EUC values minus, if any, the escape character prefix.

The relationship between upper- and lower-case letters (**ul**) is expressed as ordered pairs of octal or hexadecimal constants: *<upper-case_character lower-case_character>*. One or more space characters may separate these two constants. Zero or more space may separate angle brackets (< >) from numbers.

The following is the format of an input specification for **cswidth**: *n1:s1,n2:s2,n3:s3* where:

- n1* byte width for supplementary code set 1, required
- s1* screen width for supplementary code set 1
- n2* byte width for supplementary code set 2
- s2* screen width for supplementary code set 2
- n3* byte width for supplementary code set 3
- s3* screen width for supplementary code set 3

decimal_point and **thousands_sep** are specified by a single character, the delimiter.

EXAMPLES

Here is the input file used to create the `iso_8859_1` codeset definition table.

```

LC_CTYPE      LC_CTYPE
isupper       0x41 - 0x5a      0xc0 - 0xd6      0xd8 - 0xde
islower       0x61 - 0x7a      0xdf             0xe0 - 0xf6      0xf8 - 0xff
isdigit       0x30 - 0x39
isspace       0x20
ispunct       0x21 - 0x2f      0x3a - 0x40      0x5b - 0x60      0x7b - 0x7e \
              0xa1 - 0xbf      0xd7             0xf7
iscntrl       0x0 - 0x1f      0x7f
isblank       0x20
isxdigit      0x30 - 0x39      0x61 - 0x66      0x41 - 0x46
ul            <0x41 0x61>      <0x42 0x62>      <0x43 0x63>      <0x44 0x64> \
              <0x45 0x65>      <0x46 0x66>      <0x47 0x67>      <0x48 0x68> \
              <0x49 0x69>      <0x4a 0x6a>      <0x4b 0x6b>      <0x4c 0x6c> \
              <0x4d 0x6d>      <0x4e 0x6e>      <0x4f 0x6f>      <0x50 0x70> \
              <0x51 0x71>      <0x52 0x72>      <0x53 0x73>      <0x54 0x74> \
              <0x55 0x75>      <0x56 0x76>      <0x57 0x77>      <0x58 0x78> \
              <0x59 0x79>      <0x5a 0x7a>      <0xc0 0xe0>      <0xc1 0xe1> \
              <0xc2 0xe2>      <0xc3 0xe3>      <0xc4 0xe4>      <0xc5 0xe5> \
              <0xc6 0xe6>      <0xc7 0xe7>      <0xc8 0xe8>      <0xc9 0xe9> \
              <0xca 0xea>      <0xcb 0xeb>      <0xcc 0xec>      <0xcd 0xed> \
              <0xce 0xee>      <0xcf 0xef>      <0xd0 0xf0>      <0xd1 0xf1> \
              <0xd2 0xf2>      <0xd3 0xf3>      <0xd4 0xf4>      <0xd5 0xf5> \
              <0xd6 0xf6>      <0xd8 0xf8>      <0xd9 0xf9>      <0xda 0xfa> \
              <0xdb 0xfb>      <0xdc 0xfc>      <0xdd 0xfd>      <0xde 0xfe>

cswidth       1:1,0:0,0:0
LC_NUMERIC    LC_NUMERIC
decimal_point  " "
thousands_sep " "
#
LC_CTYPE1
isupper       0xc0 - 0xd6      0xd8 - 0xde
islower       0xdf             0xe0 - 0xf6      0xf8 - 0xff
isspace       0xa0
ispunct       0xa1 - 0xbf      0xd7             0xf7
isblank       0xa0
ul            <0xc0 0xe0>      <0xc1 0xe1> \
              <0xc2 0xe2>      <0xc3 0xe3>      <0xc4 0xe4>      <0xc5 0xe5> \
              <0xc6 0xe6>      <0xc7 0xe7>      <0xc8 0xe8>      <0xc9 0xe9> \
              <0xca 0xea>      <0xcb 0xeb>      <0xcc 0xec>      <0xcd 0xed> \
              <0xce 0xee>      <0xcf 0xef>      <0xd0 0xf0>      <0xd1 0xf1> \
              <0xd2 0xf2>      <0xd3 0xf3>      <0xd4 0xf4>      <0xd5 0xf5> \
              <0xd6 0xf6>      <0xd8 0xf8>      <0xd9 0xf9>      <0xda 0xfa> \
              <0xdb 0xfb>      <0xdc 0xfc>      <0xdd 0xfd>      <0xde 0xfe>

```

FILES	/usr/include/ctype.h	declarations used by character classification and conversion routines
	/usr/include/wctype.h	declarations used by wide character classification and conversion routines
	/usr/lib/locale/locale/LC_CTYPE/ctype	data file containing character classification, conversion, and codeset width information
	/usr/lib/locale/locale/LC_NUMERIC	data file containing numeric layout information
SEE ALSO	ctype(3C), setlocale(3C), iswalph(3I), environ(5)	
NOTES	Do not change files under the C locale, as this could cause undefined or nonstandard behavior.	

NAME	clri, dcopy – clear inode				
SYNOPSIS	clri [-F <i>FSType</i>] [-V] <i>special i-number</i> dcopy [-F <i>FSType</i>] [-V] <i>special i-number</i>				
AVAILABILITY	SUNWcsu				
DESCRIPTION	<p>clri writes zeros on the inodes with the decimal <i>i-number</i> on the filesystem stored on <i>special</i>. After clri, any blocks in the affected file will show up as missing in an fsck(1M) of <i>special</i>.</p> <p>Read and write permission is required on the specified file system device. The inode becomes allocatable.</p> <p>The primary purpose of this routine is to remove a file which for some reason appears in no directory. If it is used to zap an inode which does appear in a directory, care should be taken to track down the entry and remove it. Otherwise, when the inode is reallocated to some new file, the old entry will still point to that file. At that point removing the old entry will destroy the new file. The new entry will again point to an unallocated inode, so the whole cycle is likely to be repeated again and again.</p> <p>dcopy is a symbolic link to clri.</p>				
OPTIONS	<p>-F <i>FSType</i> Specify the <i>FSType</i> on which to operate. The <i>FSType</i> should either be specified here or be determinable from /etc/vfstab by matching <i>special</i> with an entry in the table, or by consulting /etc/default/fs.</p> <p>-V Echo the complete command line, but do not execute the command. The command line is generated by using the options and arguments provided by the user and adding to them information derived from /etc/vfstab. This option should be used to verify and validate the command line.</p>				
FILES	<table border="0"> <tr> <td>/etc/default/fs</td> <td>default local file system type</td> </tr> <tr> <td>/etc/vfstab</td> <td>list of default parameters for each file system</td> </tr> </table>	/etc/default/fs	default local file system type	/etc/vfstab	list of default parameters for each file system
/etc/default/fs	default local file system type				
/etc/vfstab	list of default parameters for each file system				
SEE ALSO	fsck(1M) , vfstab(4)				
NOTES	This command may not be supported for all <i>FSTypes</i> .				

NAME	colltbl – create string collation routines
SYNOPSIS	colltbl [<i>filename</i>]
DESCRIPTION	<p>The colltbl command reads locale specifications for collation order from <i>filename</i>, then creates a shared library composed of four functions: strxfrm(3C), wsxfrm(3I), strcoll(3C), and wscoll(3I). The last two transform their arguments and perform the comparison directly. If no input file is supplied, colltbl reads from standard input.</p> <p>The name of the output file is the value you assign to the keyword codeset in <i>filename</i>. The superuser should install this file as /usr/lib/locale/locale/LC_COLLATE/coll.so. It must be readable and executable by user, group, and other. Application programs consult this file when the LC_COLLATE environment is set appropriately, after having called setlocale(3C).</p> <p>The colltbl command can support languages whose collating sequence can be completely described by the following cases:</p> <ul style="list-style-type: none">• Ordering of single characters within the codeset. For example, in English B is sorted after A, but before C and D.• Equivalence class definition. A collection of characters is defined to have the same primary sort value. For example, in Finnish the letters French V and W compare equal. Both come after U but before X.• Ordering of double characters in the collation sequence. For example, in Spanish ch is collated after c, and ll is collated after l.• Ordering of one character as if it consists of two characters. For example, in German the "es-zet" ß is sorted as if it were ss. This is a special instance of the case below.• Substitution of one character string with another character string. For example, spelled-out numbers, month and day names, and so forth, can be transformed so that they sort correctly.• Null character mapping, so that certain characters in the codeset are ignored during collation. For example, if "-" were ignored during collation, then the strings re-locate and relocate would compare equal.• Secondary ordering between characters. In the case where two characters are sorted together in the collation sequence, (that is, they have the same "primary" ordering), there is sometimes a secondary ordering that is used if two strings are identical except for characters that have the same primary ordering. For example, in French, the letters e and è have the same primary ordering but e comes before è in the secondary ordering. Thus the word lever would be ordered before lèver, but lèver would be sorted before levitate. (Note that if e came before è in the primary ordering, then lèver would be sorted after levitate.)

USAGE

The specification file consists of three types of statements:

1. **codeset** *filename*

filename is the name of the output file to be created by **colltbl**.

2. **order is** *order_list*

order_list is a list of symbols, separated by semicolons, that defines the collating sequence. The special symbol ... is short-hand for symbols that are lexically sequential. For example,

```
order is    a;b;c;d;...;x;y;z
```

specifies the list of lower_case letters. Of course, this could be further shortened to **a;...;z**. Note that symbols surrounding ... must single character symbols; parentheses or braces are not allowed.

A symbol can be up to two bytes in length and can be represented in any one of the following ways:

- the symbol itself (for example, **a** for the lower-case letter **a**),
- in octal representation (for example, **\141** or **0141** for the letter **a**), or
- in hexadecimal representation (for example, **\x61** or **0x61** for the letter **a**).

Any combination of these may be used as well.

The backslash character, ****, is used for continuation. No characters are permitted after the backslash character.

Symbols enclosed in parenthesis are assigned the same primary ordering but different secondary ordering. Symbols enclosed in curly brackets are assigned only the same primary ordering. For example,

```
order is    a;b;c;ch;d;(e;è);f;...;z;\
              {1;...;9};A;...;Z
```

In the above example, **e** and **è** are assigned the same primary ordering and different secondary ordering, digits 1 through 9 are assigned the same primary ordering and no secondary ordering. Only primary ordering is assigned to the remaining symbols. Notice how double letters can be specified in the collating sequence (letter **ch** comes between **c** and **d**).

If a character is not included in the **order is** statement it is excluded from the ordering and will be ignored during sorting.

3. **substitute** *string with repl*

The **substitute** statement substitutes the string *pattern* with the string *repl*. This can be used, for example, to provide rules to sort abbreviated month names numerically:

```
substitute "Jan" with "01"
substitute "Feb" with "02"
...
substitute "Dec" with "12"
```

A simpler use of the **substitute** statement mentioned above is to substitute one character with two characters, as with the substitution of **ss** for **ß** in German.

Null character mapping can also be performed with **substitute**, as follows:

```
substitute "-" with ""
```

The **substitute** statement is optional. The **order is** and **codeset** statements are required.

Any lines in the specification file with a # in the first column are treated as comments and are ignored. Empty lines are also ignored.

EXAMPLES

The following example shows the collation specification required to support a hypothetical telephone book sorting sequence.

The sorting sequence is defined by the following rules:

- Upper and lower case letters must be sorted together, but upper case letters have precedence over lower case letters.
- All special characters and punctuation must be ignored.
- Digits must be sorted as their alphabetic counterparts (**0** as **zero**, **1** as **one**).
- The **CH**, **Ch**, **ch** combinations must be collated between **C** and **D**.
- **V** and **W**, **v** and **w** must be collated together.

The input specification file to **colltbl** should contain:

```
codeset      telephone
order is    (A;a);(B;b);(C;c);(CH;Ch;ch);(D;d);(E;e);(F;f);(G;g);\
            (H;h);(I;i);(J;j);(K;k);(L;l);(M;m);(N;n);(O;o);(P;p);\
            (Q;q);(R;r);(S;s);(T;t);(U;u);{V;W};{v;w};(X;x);(Y;y);(Z;z)

substitute "0" with "zero"
substitute "1" with "one"
substitute "2" with "two"
substitute "3" with "three"
substitute "4" with "four"
substitute "5" with "five"
substitute "6" with "six"
substitute "7" with "seven"
substitute "8" with "eight"
substitute "9" with "nine"
```

FILES

/usr/lib/locale/locale/LC_COLLATE/coll.so

shared library containing collation routines for *locale*

/opt/SUNWspro/bin/cc or any C compiler that supports these options:

- G** to output dynamically linked library
- o** to specify output filename
- O** to optimize code
- K pic** to generate position independent code

SEE ALSO

memory(3C), **setlocale(3C)**, **strcoll(3C)**, **strxfrm(3C)**, **wscoll(3I)**, **wsxfrm(3I)**, **environ(5)**

NOTES

Do not change files under the C locale, as this could cause undefined or nonstandard behavior.

NAME	convert4x – converts SunOS 4.x clients to work from Solaris 2.x servers
SYNOPSIS	<code>/usr/sbin/convert4x [export_root] [client_name]</code>
AVAILABILITY	SUNWhinst
DESCRIPTION	convert4x converts SunOS 4.x clients that were served from a SunOS 4.x server so that they will now work under the new Solaris 2.x server. The convert4x command is used after discover4x(1M) and install4x(1M) have been executed on a Solaris 2.x server that was upgraded from SunOS 4.x. It attempts to locate information about SunOS 4.x clients that were preserved through the upgrade process.
OPTIONS	<p><i>export_root</i> Specify a fully qualified pathname for the base directory of the diskless/dataless client support hierarchy. If unspecified, the default is, /export.</p> <p><i>client_name</i> Specify the name of a client that you wish to convert. If unspecified, the default is to convert all the SunOS 4.x clients that are in <i>export_root/root</i>.</p>
SEE ALSO	discover4x(1M) , install4x(1M) <i>SPARC: Installing Solaris Software</i> <i>x86: Installing Solaris Software</i>

NAME	crash – examine system images
SYNOPSIS	<code>/usr/sbin/crash [-d <i>dumpfile</i>] [-n <i>namelist</i>] [-w <i>output-file</i>]</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	The crash command is used to examine the system memory image of a running or a crashed system by formatting and printing control structures, tables, and other information. Command line arguments to crash are <i>dumpfile</i> , <i>namelist</i> , and <i>output-file</i> .
OPTIONS	<p>-d <i>dumpfile</i> Specify <i>dumpfile</i> as the file containing the system memory image. The default <i>dumpfile</i> is <code>/dev/mem</code>. The system image can also be the path-name of a dump file generated by the savecore(1M) utility.</p> <p>-n <i>namelist</i> Specify the text file <i>namelist</i> which contains the symbol table information needed for symbolic access to the system memory image to be examined. The default <i>namelist</i> is <code>/dev/ksyms</code>. Note: It is recommended that crash dumps be analyzed on a machine having the same kernel architecture as the machine that the dump was taken from.</p> <p>-w <i>output-file</i> When the crash command is invoked, a session is initiated. The output from a crash session is directed to <i>output-file</i>. The default <i>output-file</i> is the standard output.</p>
USAGE	<p>Input during a crash session is of the form:</p> <p style="padding-left: 40px;"><i>function</i> [<i>argument...</i>]</p> <p>where <i>function</i> is one of the crash functions described in the Functions subsection of this manual page, and <i>arguments</i> are qualifying data that indicate which items of the system image are to be printed.</p> <p>The default for process-related items is the current process for a running system or the process that was running at the time of the crash for a crashed system. Similarly, the default for thread-related items is the current thread for a running system or the thread that was running at the time of the crash for a crash system. If the contents of a table are being dumped, the default is all active table entries.</p>
Function Options	<p>The following function options are available to crash functions wherever they are semantically valid. Valid function options are shown in Functions.</p> <p>-e Display every entry in a table.</p> <p>-f Display the full structure.</p> <p>-p Interpret all address arguments in the command line as physical addresses. If the addresses specified are not physical addresses, results are inconsistent.</p> <p>-s <i>process</i> Specify a process slot other than the default.</p> <p>-w <i>filename</i> Redirect the output of a function to <i>filename</i>.</p>

The functions **mode**, **defproc**, and **redirect** correspond to the function options **-p**, **-s**, and **-w**. The **mode** function may be used to set the address translation mode to physical or virtual for all subsequently entered functions; **defproc** sets the value of the process slot argument for subsequent functions; and **redirect** redirects all subsequent output.

Output from **crash** functions may be piped to another program in the following way:

```
function [ argument... ] ! shell_command
```

For example,

```
mount ! grep rw
```

writes all mount table entries with an **rw** flag to the standard output. The redirection option (**-w**) cannot be used with this feature.

Depending on the context of the function, numeric arguments are assumed to be in a specific radix. Counts are assumed to be decimal. Addresses are always hexadecimal. Table address arguments larger than the size of the function table are interpreted as hexadecimal addresses; those smaller are assumed to be decimal slots in the table. Default bases on all arguments may be overridden. The C conventions for designating the bases of numbers are recognized. A number that is usually interpreted as decimal is interpreted as hexadecimal if it is preceded by **0x** and as octal if it is preceded by **0**. Decimal override is designated by **0d**, and binary by **0b**.

Aliases for functions may be any uniquely identifiable initial substring of the function name. Traditional aliases of one letter, such as **p** for **proc**, remain valid.

Many functions accept different forms of entry for the same argument. Requests for table information accept a table entry number, a physical address, a virtual address, a symbol, a range, or an expression. A range of slot numbers may be specified in the form *a-b* where *a* and *b* are decimal numbers. An expression consists of two operands and an operator. An operand may be an address, a symbol, or a number; the operator may be **+**, **-**, *****, **/**, **&**, or **|**. An operand that is a number should be preceded by a radix prefix if it is not a decimal number (**0** for octal, **0x** for hexadecimal, **0b** for binary). The expression must be enclosed in parentheses. Other functions accept any of these argument forms that are meaningful.

Two abbreviated arguments to **crash** functions are used throughout. Both accept data entered in several forms. They may be expanded into the following:

```
table_entry = slot number | address | symbol | range | expression
start_addr = address | symbol | expression
```

Functions

```
? [ -w filename ]
```

List available functions.

```
!command Escape to the shell and execute command.
```

as [**-e**] [**-f**] [**-l**] [**-p**] [**-w filename**] [*proc...*]
 Print information on process address space. If the **-l** is specified, all locks relevant to the address space structure are also displayed.

base [**-w filename**] *number...*
 Print *number* in binary, octal, decimal, and hexadecimal. A number in a radix other than decimal should be preceded by a prefix that indicates its radix as follows: **0x**, hexadecimal; **0**, octal; and **0b**, binary.

buffer [**-w filename**] [**-format**] *bufferslot*

buffer [**-w filename**] [**-format**] [**-p**] *start_addr*
 Alias: **b**
 Print the contents of a buffer in the designated format. The following *format* designations are recognized: **-b**, byte; **-c**, character; **-d**, decimal; **-x**, hexadecimal; **-o**, octal; and, **-i**, inode. If no format is given, the previous format is used. The default format at the beginning of a **crash** session is hexadecimal.

bufhdr [**-f**] [**-w filename**] [[**-p**] *table_entry...*]
 Alias: **buf**
 Print system buffer headers.

callout [**-l**] [**-w filename**]
 Alias: **c**
 Print the **callout** table. If the **-l** option is specified, the contents of the locks pertaining to the **callout** structure are also displayed.

class [**-w filename**] [*table_entry...*]
 Print information about process scheduler classes.

cpu [**-w filename**] *start_addr...*
 Display the **cpu** structure pointed to by *start_addr*.

ctx [**-w filename**] *start_addr*
 SPARC: Display the context structure given its starting address.

defproc [**-w filename**] [**-c**] [**-r**]

defproc [**-w filename**] [*slot*] [**-r**]
 Set the value of the process slot argument. The process slot argument may be set to the current process' slot number (**-c**) or the slot number may be specified. If the **-r** option is specified, the process slot number and the current thread pointer are reset (the process slot number is set to the current process' slot number and the current thread pointer is set to the currently running thread). If no argument is entered, the value of the previously set slot number is printed. At the start of a **crash** session, the process slot is set to the current process.

defthread [**-p**] [**-w filename**] [**-r**] [**-c thread_addr**]
 With no argument **defthread** displays the address of the **current thread**. If **-c** is specified, the **current thread** is set to point to the currently running thread. If an address is specified the **current thread** is specified to be that thread. If **-r** is specified, the **default process** is also set to be the process pertaining to **current thread**.

dispq [**-w filename**] [*table_entry...*]
 Print the dispatcher (scheduler) queues.

ds [**-w filename**] *virtual_address...*
 Alias: **ts**
 Print the data symbol whose address is closest to, but not greater than, the address entered.

file [**-e**] [**-f**] [**-w filename**] [[**-p**] *table_entry...*]
 Alias: **f**
 Print the **file** table.

findaddr [**-w filename**] *table slot*
 Print the address of *slot* in *table*. Only tables available to the **size** function are available to **findaddr**.

findslot [**-w filename**] *virtual_address...*
 Print the table, entry slot number, and offset for the address entered. Only tables available to the **size** function are available to **findslot**.

fs [**-f**] [**-w filename**] [[**-p**] *table_entry...*]
 Alias: **vfssw**
 Print the filename system information table.

gdp [**-e**] [**-f**] [**-l**] [**-w filename**] [[**-p**] *table_entry...*]
 x86: Print the gift descriptor protocol table. If the **-l** option is specified, display all relevant lock information.

hat [**-a**] [**-w filename**] *start_addr...*
 Display the **hat** structure pointed to by *start_addr*. If the **-a** option is specified, the chain of **hat** structures is displayed.

help [**-w filename**] *function...*
 Print a description of the named function, including syntax and aliases.

hment [**-a**] [**-w filename**] *start_addr...*
 Display the **hment** structure pointed to by *start_addr*. If the **-a** option is specified, the chain of **hment** structures is displayed.

kfp [**-w filename**] [*thread_addr*]
 Print the kernel frame pointer (**kfp**) for the start of a kernel stack trace. If *thread_addr* is specified the **kfp** for that thread is printed.

kmastat [**-w filename**]
 Print kernel memory allocator statistics.

kmausers [-e] [-f] [-w *filename*] [*cachename...*]

Print the information about the medium and large users of the kernel memory allocator that have current memory allocations. The output consists of one entry for each unique stack trace specifying the total amount of memory and number of allocations that was made with that stack trace.

This function is only available if the kernel has the **KMF_AUDIT** flag set in **kmem_flags**. (See the **NOTES** below.)

If one or more cache names (e.g. *kmem_alloc_256*) are specified the scan of memory usage is restricted to those caches. By default all caches are included.

If the (-e) option is used, the small users of the allocator are included. The small users are allocations that total less than 1024 bytes of memory or for which there are less than 10 allocations with the same stack trace.

If the (-f) option is used, the stack traces are printed for each individual allocation.

lck [-e] [-w *filename*] [[-p] *lock_addr...*]

Alias: **l**

Print record locking information. If the -e option is used or lock address arguments are given, the record lock list is printed. If no argument is entered, information on locks relative to UFS inodes is printed.

linkblk [-e] [-w *filename*] [[-p] *table_entry...*]

Print the **linkblk** table.

lwp [-w *filename*] [*lwp_addr...*]

Display the **lwp** structure pointed to by *lwp_addr*. If no address is specified as an argument, the **lwp** structure associated with the **current thread** is displayed.

major [-w *filename*] [*entry...*]

Print the **MAJOR** table.

map [-w *filename*] *mapname...*

Print the **map** structure of the given *mapname*.

mblk [-e] [-f] [-w *filename*] [[-p] *table_entry...*]

Print allocated streams message block and data block headers.

mblkusers [-e] [[-f] [-w *filename*]]

Print the information about the medium and large users of the stream message memory allocator that have current mblk allocations. The output consists of one entry for each unique stack trace specifying the total amount of memory and number of allocations that was made with that stack trace.

This function is only available if the kernel has the **KMF_AUDIT** flag set in **kmem_flags**. (See the **NOTES** below.)

If the (-e) option is used, the small users of the allocator are included. The small users are allocations that total less than 1024 bytes of memory or for which there are less than 10 allocations with the same stack trace.

If the **(-f)** option is used, the stack traces are printed for each individual allocation.

mode [**-w filename**] [*mode*]

Set address translation of arguments to virtual (**v**) or physical (**p**) mode. If no mode argument is given, the current mode is printed. At the start of a **crash** session, the mode is virtual.

mount [**-f**] [**-w filename**] [[**-p**] *table_entry...*]

Alias: **m, vfs**

Print information about mounted filename systems.

mutex [**-w filename**] *mutex_addr...*

Dump the contents of the **mutex** pointed to by *mutex_addr*.

mutexable [**-w filename**]

Display every entry in the **mutex_init_table**.

nfsnode [**-w filename**] [*node_addr*]

Print the NFS remote nodes.

nm [**-w filename**] *symbol...*

Print value and type for the given symbol.

od [**-p**] [**-w filename**] [*-format*] [*-mode*] [**-s process**] *start_addr* [*count*]

Alias: **rd**

Print *count* values starting at *start_addr* in one of the following formats: character (**-c**), decimal (**-d**), hexadecimal (**-x**), octal (**-o**), ASCII (**-a**), or hexadecimal/character (**-h**), and one of the following modes: long (**-l**), short (**-t**), or byte (**-b**). The default mode for character and ASCII formats is byte; the default mode for decimal, hexadecimal, and octal formats is long. The format **-h** prints both hexadecimal and character representations of the addresses dumped; no mode needs to be specified. When format or mode is omitted, the previous value is used. At the start of a **crash** session, the format is hexadecimal and the mode is long. If no count is entered, 1 is assumed.

page [**-e**] [**-l**] [**-w filename**] [[**-p**] *table_entry...*]

Print information about pages. If **-l** is specified, display all relevant locking information.

pcb [**-w filename**] [*thread_addr*]

Print the process control block. If no arguments are given, the active **pcb** for the current thread is printed. If *thread_addr* is specified, the **pcb** associated with that thread is printed.

pcfsnode [**-w filename**] [*node_address*]

Print the PCFS filename and directory node tables.

pmgrp [**-l**] [**-w filename**] *start_addr...*

Display the **pmgrp** structure pointed to by *start_addr*. If the **-l** option is specified, the chain of **pmgrp** structures is displayed.

prnode [-e] [-f] [-l] [-w *filename*] [[-p] *table_entry...*]
 Print information about the private data of processes being traced. If -l is specified, all relevant locking information is also displayed.

proc [-e] [-f] [-l] [-w *filename*] [[-p] [-a] *table_entry...* | #*procid...*]

proc [-e] [-f] [-l] [-w *filename*] [-r]
 Alias: **p**
 Print the process table. Process table information may be specified in two ways. First, any mixture of table entries and process IDs may be entered. Each process ID must be preceded by a #. Alternatively, process table information for runnable processes may be specified with the runnable option (-r). If the -l option is specified, all relevant locking information is displayed.

pte [-w *filename*] *start_addr...*
 Display the **pte** structure pointed to by *start_addr*.

pty [-e] [-f] [-w *filename*] [-s] [-h] [-l] [-t *type*] [[-p] *table_entry*]
 Print the pseudo ttys presently configured. The -l, -h and -s options give information about the STREAMS modules **ldterm**, **psem** and **pckt**, respectively. If the -t option is used, the table for the single **pty** type specified is printed.

qrun [-w *filename*]
 Print the list of scheduled STREAMS queues.

queue [-e] [-f] [-w *filename*] [[-p] *table_entry...*]
 Print the allocated STREAMS queues.

quit Alias: **q**
 Terminate the **crash** session. Exit.

redirect [-w *filename*] [-c]

redirect [-w *filename*] [*newfile*]
 Used with a *filename*, redirects output of a **crash** session to *newfile*. If no argument is given, the filename name to which output is being redirected is printed. Alternatively, the close option (-c) closes the previously set filename and redirects output to the standard output.

rtdptbl [-w *filename*] [*table_entry...*]
 Print the real-time dispatcher (scheduler) parameter table. See **rt_dptbl**(4).

rtproc [-w *filename*]
 Print information about processes in the real-time scheduler class.

rwlock [-w *filename*] *rw_addr...*
 Dump the contents of the **rwlock** (read/write lock) structure pointed to by *rw_addr*.

search [-w *filename*] [-m *mask*] [-s *process*] *pattern* [-p] *start_addr length*
 Print the words in memory that match *pattern*, beginning at the *start_addr* for *length* words. The mask is AND'ed (&) with each memory word and the result compared against the pattern. The mask defaults to **0xffffffff**.

sema [*-w filename*] *sema_addr...*
 Dump the contents of the semaphore structure pointed to by *sema_addr*.

size [*-x*] [*-w filename*] [*structure_name...*]
 Print the size of the designated structure. The *-x* option prints the size in hexadecimal. If no argument is given, a list of the structure names for which sizes are available is printed. The following example lists those structures:

```
> size
```

buf	callout	flckinfo	fifonode
filock	flox	lwp	pp
prnode	proc	snode	thread
tty	ufs_inode	user	vfs
vfssw	vnode		
datab	dblk	linkblk	
mblk	msgb	queue	
stdata	streams		

```
>
```

sment [*-w filename*] *start_addr...*
 Display the **sment** structure pointed to by *start_addr*.

smgrp [*-l*] [*-w filename*] *start_addr...*
 Display the **smgrp** structure pointed to by *start_addr*. If the *-l* option is specified, the chain of **smgrp** structures is displayed.

snode [*-e*] [*-f*] [*-l*] [*-w filename*] [[*-p*] *table_entry...*]
 Print information about open special filenames. If the *-l* option is specified, all relevant locking information is also displayed.

stack [*-w filename*] [*-u*] [*-p*] [*thread*]
stack [*-w filename*] [*-k*] [*-p*] [*thread*]
 Alias: **s**
 Dump the stack. The *-u* option prints the user stack. The *-k* option prints the kernel stack. If no arguments are entered, the kernel stack for the current thread is printed. Otherwise, the kernel stack for the currently running thread is printed.

status [*-w filename*]
 Print system statistics.

stream [*-e*] [*-f*] [*-w filename*] [[*-p*] *start_addr...*]
 Print the STREAMS table.

strstat [*-w filename*]
 Print STREAMS statistics.

trace [*-w filename*] [[*-p*] [*thread_addr*]]
 Alias: **t**
 Print stack trace. Displays the stack trace of the currently running or the specified thread.

thread [-e] [-f] [-l] [-w *filename*] *slot number*
 With no option, displays the **current thread**. If -e is specified, all threads in the system are displayed. Otherwise all threads pointed to by the addresses specified as argument are displayed.

ts [-w *filename*] *virtual_address...*
 Alias: **ds**
 Print text symbol closest to the designated address.

tsdptbl [-w *filename*] [*table_entry...*]
 Print the time-sharing dispatcher parameter table. See **ts_dptbl**(4).

tsproc [-w *filename*]
 Print information about processes in the time-sharing dispatcher parameter table (scheduler class).

tty [-e] [-f] [-l] [-w *filename*] [-t *type* [[-p] *table_entry...*]]

tty [-e] [-f] [-l] [-w *filename*] [[-p] *start_addr*]
 Valid types: **pp, iu**.
 Print the tty table. If no arguments are given, the tty table for both tty types is printed. If the -t option is used, the table for the single tty type specified is printed. If no argument follows the type option, all entries in the table are printed. A single tty entry may be specified using *start_addr*. The -l option prints the line discipline information.

uinode [-d] [-e] [-f] [-l] [-r] [-w *filename*] [[-p] *table_entry...*]
 Alias: **ui**
 Print the UFS inode table. The -d option will list the address and i-number of all UFS inodes in use and on the free list. If the -l option is specified, all relevant locking information is also displayed. The -r option will display all free UFS inodes.

user [-e] [-f] [-l] [-w *filename*] [*process*]
 Alias: **u**
 Print the user structure for the designated process. If the -l option is specified, display all relevant locking information.

var [-w *filename*]
 Alias: **v**
 Print the tunable system parameters.

vfs [-e] [-w *filename*] [[-p] *address...*]
 Alias: **m, mount**
 Print information about mounted filename systems.

vfssw [-f] [-w *filename*] [[-p] *table_entry...*]
 Alias: **fs**
 Print information about configured filename system types.

vnnode [-w *filename*] [-l] [-p] *vnnode_addr...*
 Print information about vnodes.

vtop [*-w filename*] [*-s process*] *start_addr...*

Print the physical address translation of the virtual address *start_addr*.

EXIT STATUS

The following exit values are returned:

0 Success.
1 An error has occurred.

FILES

/dev/mem system image of currently running system
/dev/ksyms system namelist

SEE ALSO

savecore(1M), **ts_dptbl(4)**

NOTES

Kernel core dumps should be examined on the same platform they were created on.

The following steps are necessary in order to set **KMF_AUDIT** in **kmem_flags** (which is needed for the **kmausers** and **mblkusers** commands):

```
ok boot kadb -d
kadb: kernel/unix
kadb[0]: startup:b
kadb[0]: :c
kadb[0]: kmem_flags/W1
kadb[0]: :c
```

NAME	cron – clock daemon	
SYNOPSIS	<code>/usr/sbin/cron</code>	
AVAILABILITY	SUNWcsu	
DESCRIPTION	<p>The cron command starts a process that executes commands at specified dates and times. Regularly scheduled commands can be specified according to instructions found in crontab files in the directory <code>/var/spool/cron/crontabs</code>. Users can submit their own crontab file using the crontab(1) command. Commands which are to be executed only once may be submitted using the at(1) command.</p> <p>cron only examines crontab or at command files during its own process initialization phase and when the crontab or at command is run. This reduces the overhead of checking for new or changed files at regularly scheduled intervals.</p> <p>Since cron never exits, it should be executed only once. This is done routinely through <code>/etc/rc2.d/S75cron</code> at system boot time. The file <code>/etc/cron.d/FIFO</code> is used (among other things) as a lock file to prevent the execution of more than one instance of cron.</p> <p>cron captures the output of the job's stdout and stderr streams, and, if it is non-empty, mails the output to the user. If the job does not produce output, no mail is sent to the user (unless the job is an at(1) job and the -m option was specified when the job was submitted).</p>	
Setting cron Defaults	<p>To keep a log of all actions taken by cron, CRONLOG=YES (by default) must be specified in the <code>/etc/default/cron</code> file. If CRONLOG=NO is specified, no logging is done. Keeping the log is a user configurable option since cron usually creates huge log files.</p> <p>The PATH for user cron jobs can be set using PATH= in <code>/etc/default/cron</code>. The PATH for root cron jobs can be set using SUPATH= in <code>/etc/default/cron</code>. The security implications of setting PATH and SUPATH should be carefully considered.</p> <p>Example <code>/etc/default/cron</code> file:</p> <pre> CRONLOG=YES PATH=/usr/bin:/usr/ucb: </pre> <p>This example enables logging and sets the default PATH used by non-root jobs to <code>/usr/bin:/usr/ucb:</code>. Root jobs will continue to use <code>/usr/sbin:/usr/bin</code>.</p> <p><code>/etc/cron.d/logchecker</code> is a script that checks to see if the log file has exceeded the system ulimit. If so, the log file is moved to <code>/var/cron/olog</code>.</p>	
FILES	<code>/etc/cron.d</code> <code>/etc/cron.d/FIFO</code> <code>/etc/default/cron</code> <code>/var/cron/log</code> <code>/var/spool/cron</code> <code>/etc/cron.d/logchecker</code>	main cron directory used as a lock file contains cron default settings cron history information spool area moves log file to <code>/var/cron/olog</code> if log file exceeds system ulimit .

/etc/cron.d/queuedefs queue description file for **at**, **batch**, and **cron**.

SEE ALSO

at(1), **crontab(1)**, **sh(1)**, **queuedefs(4)**

DIAGNOSTICS

A history of all actions taken by **cron** is stored in **/var/cron/log** and (possibly) **/var/cron/olog**.

NAME	dd – convert and copy a file
SYNOPSIS	dd [<i>operand=value ...</i>]
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>dd copies the specified input file to the specified output with possible conversions. The standard input and output are used by default. The input and output block sizes may be specified to take advantage of raw physical I/O. Sizes are specified in bytes; a number may end with k, b, or w to specify multiplication by 1024, 512, or 2, respectively. Numbers may also be separated by x to indicate multiplication.</p> <p>dd will read the input one block at a time, using the specified input block size; it then will process the block of data actually returned, which could be smaller than the requested block size. dd will apply any conversions that have been specified and write the resulting data to the output in blocks of the specified output block size.</p> <p>cbs is used only if ascii, unblock, ebcdic, ibm, or block conversion is specified. In the first two cases, cbs characters are copied into the conversion buffer, any specified character mapping is done, trailing blanks are trimmed, and a new-line is added before sending the line to output. In the last three cases, characters up to new-line are read into the conversion buffer and blanks are added to make up an output record of size cbs. ASCII files are presumed to contain new-line characters. If cbs is unspecified or 0, the ascii, ebcdic, and ibm options convert the character set without changing the input file's block structure; the unblock and block options become a simple file copy.</p> <p>After completion, dd reports the number of whole and partial input and output blocks.</p>
OPERANDS	<p>The following operands are supported:</p> <p>if=file Specify the input path; standard input is the default.</p> <p>of=file Specify the output path; standard output is the default. If the seek=expr conversion is not also specified, the output file will be truncated before the copy begins, unless conv=notrunc is specified. If seek=expr is specified, but conv=notrunc is not, the effect of the copy will be to preserve the blocks in the output file over which dd seeks, but no other portion of the output file will be preserved. (If the size of the seek plus the size of the input file is less than the previous size of the output file, the output file will be shortened by the copy.)</p> <p>ibs=n Specify the input block size in <i>n</i> bytes (default is 512).</p> <p>obs=n Specify the output block size in <i>n</i> bytes (default is 512).</p> <p>bs=n Set both input and output block sizes to <i>n</i> bytes, superseding ibs= and obs=. If no conversion other than sync, noerror, and notrunc is specified, each input block will be copied to the output as a single block without aggregating short blocks.</p> <p>cbs=n Specify the conversion block size for block and unblock in bytes by <i>n</i></p>

(default is **0**). If **cbs=** is omitted or given a value of **0**, using **block** or **unblock** produces unspecified results.

This option is used only if ASCII or EBCDIC conversion is specified. For the **ascii** operand, the input is handled as described for the **unblock** operand except that characters are converted to ASCII before the trailing SPACE characters are deleted. For the **ebcdic** and **ibm** operands, the input is handled as described for the **block** operand except that the characters are converted to EBCDIC or IBM EBCDIC after the trailing SPACE characters are added.

files=<i>n</i>	Copy and concatenate <i>n</i> input files before terminating (makes sense only where input is a magnetic tape or similar device).
skip=<i>n</i>	Skip <i>n</i> input blocks (using the specified input block size) before starting to copy. On seekable files, the implementation will read the blocks or seek past them; on non-seekable files, the blocks will be read and the data will be discarded.
iseek=<i>n</i>	Seek <i>n</i> blocks from beginning of input file before copying (appropriate for disk files, where skip can be incredibly slow).
oseek=<i>n</i>	Seek <i>n</i> blocks from beginning of output file before copying.
seek=<i>n</i>	Skip <i>n</i> blocks (using the specified output block size) from beginning of output file before copying. On non-seekable files, existing blocks will be read and space from the current end-of-file to the specified offset, if any, filled with null bytes; on seekable files, the implementation will seek to the specified offset or read the blocks as described for non-seekable files.
count=<i>n</i>	Copy only <i>n</i> input blocks.
conv=<i>value</i>[,<i>value</i>...]	

Where *values* are comma-separated symbols from the following list.

ascii	Convert EBCDIC to ASCII.
ebcdic	Convert ASCII to EBCDIC. If converting fixed-length ASCII records without NEWLINES, set up a pipeline with dd conv=unblock beforehand.
ibm	Slightly different map of ASCII to EBCDIC. If converting fixed-length ASCII records without NEWLINES, set up a pipeline with dd conv=unblock beforehand.

The **ascii**, **ebcdic**, and **ibm** values are mutually exclusive.

block Treat the input as a sequence of NEWLINE-terminated or EOF-terminated variable-length records independent of the input block boundaries. Each record is converted to a record with a fixed length specified by the conversion block size. Any NEWLINE character is removed from the input line; SPACE characters are appended to lines that are shorter than their conversion block size to fill the block. Lines that are

longer than the conversion block size are truncated to the largest number of characters that will fit into that size; the number of truncated lines is reported.

unblock Convert fixed-length records to variable length. Read a number of bytes equal to the conversion block size (or the number of bytes remaining in the input, if less than the conversion block size), delete all trailing SPACE characters, and append a NEWLINE character.

The **block** and **unblock** values are mutually exclusive.

lcase Map upper-case characters specified by the **LC_CTYPE** keyword **tolower** to the corresponding lower-case character. Characters for which no mapping is specified will not be modified by this conversion.

ucase Map lower-case characters specified by the **LC_CTYPE** keyword **toupper** to the corresponding upper-case character. Characters for which no mapping is specified will not be modified by this conversion.

The **lcase** and **ucase** symbols are mutually exclusive.

swab Swap every pair of input bytes. If the current input record is an odd number of bytes, the last byte in the input record is ignored.

noerror Do not stop processing on an input error. When an input error occurs, a diagnostic message will be written on standard error, followed by the current input and output block counts in the same format as used at completion. If the **sync** conversion is specified, the missing input will be replaced with null bytes and processed normally; otherwise, the input block will be omitted from the output.

notrunc Do not truncate the output file. Preserve blocks in the output file not explicitly written by this invocation of **dd**. (See also the preceding **of=file** operand.)

sync Pad every input block to the size of the **ibs=** buffer, appending null bytes. (If either **block** or **unblock** is also specified, append SPACE characters, rather than null bytes.)

If operands other than **conv=** are specified more than once, the last specified **operand=value** will be used.

For the **bs=**, **cbs=**, **ibs=**, and **obs=** operands, the application must supply an expression specifying a size in bytes. The expression, *expr*, can be:

1. a positive decimal number
2. a positive decimal number followed by **k**, specifying multiplication by 1024
3. a positive decimal number followed by **b**, specifying multiplication by 512

- two or more positive decimal numbers (with or without **k** or **b**) separated by **x**, specifying the product of the indicated values.

All of the operands will be processed before any input is read.

EXAMPLES

- The following command:

```
example% dd if=/dev/rmt/0h of=/dev/rmt/1h
```

copies from tape drive 0 to tape drive 1, using a common historical device naming convention.

- The following command:

```
example% dd ibs=10 skip=1
```

strips the first 10 bytes from standard input.

- This example reads an EBCDIC tape blocked ten 80-byte EBCDIC card images per block into the ASCII file **x**:

```
example% dd if=/dev/tape of=x ibs=800 cbs=80 conv=ascii,lcase
```

- Use **conv=sync** when writing to a tape as in the following example:

```
example% tar cvf - . | compress | dd obs=1024k of=/dev/rmt/0 conv=sync
```

ENVIRONMENT

See **environ(5)** for descriptions of the following environment variables that affect the execution of **dd**: **LC_CTYPE**, **LC_MESSAGES**, and **NLSPATH**.

EXIT STATUS

The following exit values are returned:

- 0** The input file was copied successfully.
- >0** An error occurred.

If an input error is detected and the **noerror** conversion has not been specified, any partial output block will be written to the output file, a diagnostic message will be written, and the copy operation will be discontinued. If some other error is detected, a diagnostic message will be written and the copy operation will be discontinued.

SEE ALSO

cp(1), **sed(1)**, **tr(1)**, **environ(5)**

DIAGNOSTICS

f+p records in(out) numbers of full and partial blocks read(written)

NOTES

Do not use **dd** to copy files between file systems having different block sizes.

Using a blocked device to copy a file will result in extra nulls being added to the file to pad the final block to the block boundary.

When **dd** reads from a pipe, using the **ibs=X** and **obs=Y** operands, the output will always be blocked in chunks of size Y. When **bs=Z** is used, the output blocks will be whatever was available to be read from the pipe at the time.

When using **dd** to copy files to a tape device, the file size must be a multiple of the device sector size (for example, 512 Kbyte). To copy files of arbitrary size to a tape device, use **tar(1)** or **cpio(1)**.

For **SIGINT**, **dd** will write status information to standard error before exiting. It will take the standard action for all other signals.

NAME	deallocate – device deallocation								
SYNOPSIS	deallocate [-s] <i>device</i> deallocate [-s] [-F] <i>device</i> deallocate [-s] -I								
AVAILABILITY	The functionality described in this man page is available only if the Basic Security Module (BSM) has been enabled. See bsmconv(1M) for more information.								
DESCRIPTION	<p>deallocate deallocates a <i>device</i> allocated to the evoking user. <i>device</i> can be a device defined in device_allocate(4) or one of the device special files associated with the device. It resets the ownership and the permission on all device special files associated with <i>device</i>, disabling the user's access to that device. This option can be used by the super user to remove access to the device by another user.</p> <p>When deallocation or forced deallocation is performed, the appropriate device cleaning program is executed, based on the contents of device_allocate(4). These cleaning programs are normally stored in /etc/security/lib.</p>								
OPTIONS	<table border="0"> <tr> <td style="padding-right: 20px;"><i>device</i></td> <td>Deallocate the device associated with the device special file specified by <i>device</i>.</td> </tr> <tr> <td>-s</td> <td>Silent. Suppress any diagnostic output.</td> </tr> <tr> <td>-F <i>device</i></td> <td>Force deallocation of the device associated with the file specified by <i>device</i>. Only the super user is permitted to use this option.</td> </tr> <tr> <td>-I</td> <td>Force deallocation of all allocatable devices. Only the super user is permitted to use this option. This option should only be used at system initialization.</td> </tr> </table>	<i>device</i>	Deallocate the device associated with the device special file specified by <i>device</i> .	-s	Silent. Suppress any diagnostic output.	-F <i>device</i>	Force deallocation of the device associated with the file specified by <i>device</i> . Only the super user is permitted to use this option.	-I	Force deallocation of all allocatable devices. Only the super user is permitted to use this option. This option should only be used at system initialization.
<i>device</i>	Deallocate the device associated with the device special file specified by <i>device</i> .								
-s	Silent. Suppress any diagnostic output.								
-F <i>device</i>	Force deallocation of the device associated with the file specified by <i>device</i> . Only the super user is permitted to use this option.								
-I	Force deallocation of all allocatable devices. Only the super user is permitted to use this option. This option should only be used at system initialization.								
DIAGNOSTICS	deallocate returns a nonzero exit status in the event of an error.								
FILES	/etc/security/device_allocate /etc/security/device_maps /etc/security/dev/* /etc/security/lib/*								
SEE ALSO	allocate(1M) , bsmconv(1M) , device_allocate(4) , device_maps(4)								

NAME	devattr – lists device attributes
SYNOPSIS	devattr [-v] <i>device</i> [<i>attribute ...</i>]
DESCRIPTION	devattr displays the values for a device's attributes. The display can be presented in two formats. Used without the -v option, only the attribute values are shown. Used with the -v option, the attributes are shown in an <i>attribute=value</i> format. When no attributes are given on the command line, all attributes for the specified device are displayed in alphabetical order by attribute name. If attributes are given on the command line, only those attributes are shown, displayed in command line order.
OPTIONS	-v Specifies verbose format. Attribute values are displayed in an <i>attribute=value</i> format. <i>device</i> Defines the device whose attributes should be displayed. Can be the pathname of the device or the device alias. <i>attribute</i> Defines which attribute, or attributes, should be shown. Default is to show all attributes for a device. See the putdev(1M) manual page for a complete listing and description of available attributes.
ERRORS	The command will exit with one of the following values: 0 = successful completion of the task. 1 = command syntax incorrect, invalid option used, or internal error occurred. 2 = device table could not be opened for reading. 3 = requested device could not be found in the device table. 4 = requested attribute not defined for specified device.
FILES	/etc/device.tab
SEE ALSO	getdev(1M) , putdev(1M)

NAME	devconfig – configure device attributes
SYNOPSIS	devconfig
AVAILABILITY	x86 SUNWinst
DESCRIPTION	<p>devconfig is an interactive editor for device driver hardware configuration files and the OpenWindows configuration file.</p> <p>Devices that are not self-identifying require that correct information be supplied in the hardware configuration file in order for the device to be recognized. Devconfig is designed to ease the task of maintaining correct device information in the configuration files.</p> <p>Prototype configuration information files stored in /usr/lib/devconfig are used to check user input to ensure that the values provided for each attribute are of the correct type and fall within legal bounds. See device.cinfo(4) for a description of the format of configuration information files. The location for the cinfo files can be set by the DEVCONFIGHOME environment variable.</p> <p>After making changes to a hardware configuration file that has a driver associated with it, an attempt is made to reload the driver to verify the attributes. Some drivers may not be unloadable. In this case, a system reboot must be initiated before the new attributes can take effect. If necessary, devconfig also updates the OpenWindows configuration file, OWconfig (see the <i>OpenWindows Desktop Reference Manual</i>). devconfig makes a backup copy of a modified file in a .bak file. In addition, the first version of OWconfig is saved in OWconfig.save. This is because the original version of OWconfig contains helpful prototype information that may be referred to in case OWconfig needs to be edited manually.</p> <p>If the default location for configuration files is not writable (as is the case during installation) devconfig writes the updated files in the same location relative to the directory /tmp/root. No attempt is made to reload the driver in this case.</p> <p>devconfig is controlled by a simple menu system. The Up/Down arrow keys move the cursor to different items in a menu. The Left/Right arrow keys move the cursor to different items in a field. The Enter key selects an item. (Note that the Enter key may be labeled Return on some keyboards.) See the online help for more guidance.</p>
Operation	<p>devconfig first displays a list of configured devices in the system. Selecting a configured device allows you to view its attributes or unconfigure it. Self-identifying devices can not be unconfigured by devconfig.</p> <p>When you add a new device, devconfig displays the supported device categories. After choosing a device category, devconfig displays the devices supported in that category. Self-identifying devices cannot be added with devconfig and they are not displayed in the list of the devices. After you have selected the device to be added, devconfig displays the list of the device attributes.</p>

Once you have chosen the proper values for the attributes and applied them by using the Apply button, the device is added to the list of configured devices. You may cancel an operation by using the Cancel button.

FILES	/kernel/drv/*.conf	hardware configuration files
	/usr/lib/devconfig/*.cinfo	configuration information files
	/usr/openwin/server/etc/OWconfig	network OpenWindows configuration file
	/etc/openwin/server/etc/OWconfig	local OpenWindows configuration file

SEE ALSO **drvconfig(1M)**, **prtconf(1M)**, **device.cinfo(4)**,
OpenWindows Desktop Reference Manual

NAME	devfree – release devices from exclusive use
SYNOPSIS	devfree <i>key</i> [<i>device</i> ...]
DESCRIPTION	devfree releases devices from exclusive use. Exclusive use is requested with the command devreserv . When devfree is invoked with only the <i>key</i> argument, it releases all devices that have been reserved for that <i>key</i> . When called with <i>key</i> and <i>device</i> arguments, devfree releases the specified devices that have been reserved with that <i>key</i> .
OPTIONS	<i>key</i> Designates the unique key on which the device was reserved. <i>device</i> Defines device that this command will release from exclusive use. Can be the pathname of the device or the device alias.
ERRORS	The command will exit with one of the following values: 0 Successful completion of the task. 1 Command syntax incorrect, invalid option used, or internal error occurred. 2 Device table or device reservation table could not be opened for reading. 3 Reservation release could not be completely fulfilled because one or more of the devices was not reserved or was not reserved on the specified key.
FILES	<i>/etc/device.tab</i> <i>/etc/devlkfile</i>
SEE ALSO	devreserv (1M)
NOTES	The commands devreserv and devfree are used to manage the availability of devices on a system. These commands do not place any constraints on the access to the device. They serve only as a centralized bookkeeping point for those who wish to use them. Processes that do not use devreserv may concurrently use a device with a process that has reserved that device.

NAME	devinfo – print device specific information
SYNOPSIS	<code>/usr/sbin/devinfo -p device -i device</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	The devinfo command is used to print device specific information about disk devices on standard out. The command can only be used by the super-user.
OPTIONS	<p>-i Prints the following device information:</p> <ul style="list-style-type: none">Device nameSoftware version (not supported and prints as 0 in Solaris 2.x)Drive id number (not supported and prints as 0 in Solaris 2.x)Device blocks per cylinderDevice bytes per blockNumber of device partitions with a block size greater than zero <p>-p Prints the following device partition information:</p> <ul style="list-style-type: none">Device nameDevice major and minor numbers (in hexadecimal)Partition start blockNumber of blocks allocated to the partitionPartition flagPartition tag <p>This command is used by various other commands to obtain device specific information for the making of file systems and determining partition information.</p>
SEE ALSO	prtvtoc(1M)

NAME	devlinks – adds /dev entries for miscellaneous devices and pseudo-devices
SYNOPSIS	/usr/sbin/devlinks [-d] [-r <i>rootdir</i>] [-t <i>table-file</i>]
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>devlinks creates symbolic links from the /dev directory tree to the actual block- and character-special device nodes under the /devices directory tree. The links are created according to specifications found in the <i>table-file</i> (by default /etc/devlink.tab).</p> <p>devlinks is called each time the system is reconfiguration-booted, and can only be run after drvconfig(1M) is run, since drvconfig(1M) builds the kernel data structures and the /devices tree.</p> <p>The <i>table-file</i> (normally /etc/devlink.tab) is an ASCII file, with one line per record. Comment lines, which must contain a hash character ('#') as their first character, are allowed. Each entry must contain at least two fields, but may contain three fields. Fields are separated by single TAB characters.</p> <p>The fields are:</p> <p><i>devfs-spec</i> Specification of devinfo nodes that will have links created for them. This specification consists of one or more keyword-value pairs, where the keyword is separated from the value by an equal-sign ('='), and keyword-value pairs are separated from one another by semicolons.</p> <p>The possible keywords are:</p> <p><i>type</i> The devinfo device type. Possible values are specified in ddi_create_minor_node(9F).</p> <p><i>name</i> The name of the node. This is the portion of the /devices tree entry name that occurs before the first '@' or ':' character.</p> <p><i>addr</i>[<i>n</i>] The address portion of a node name. This is the portion of a node name that occurs between the '@' and the ':' characters. It is possible that a node may have a name without an address part, which is the case for many of the pseudo-device nodes. If a number is given after the <i>addr</i> it specifies a match of a particular comma-separated subfield of the address field: <i>addr1</i> matches the first subfield, <i>addr2</i> matches the second, and so on. <i>addr0</i> is the same as <i>addr</i> and matches the whole field.</p> <p><i>minor</i>[<i>n</i>] The minor portion of a node name – the portion of the name after the ':'. As with <i>addr</i> above, a number after the <i>minor</i> keyword specifies a subfield to match.</p> <p>Of these four specifications, only the <i>type</i> specification must always be present.</p>

<i>name</i>	<p>Specification of the /dev links that correspond to the devinfo nodes. This field allows devlinks to determine matching /dev names for the /devices nodes it has found. The specification of this field uses escape-sequences to allow portions of the /devices name to be included in the /dev name, <i>or</i> to allow a counter to be used in creating node names. If a counter is used to create a name, the portion of the name before the counter must be specified absolutely, and all names in the /dev/- subdirectory that match (up to and including the counter) are considered to be subdevices of the same device. This means that they should all point to the same directory, name and address under the /devices/-tree</p> <p>The possible escape-sequences are:</p> <p>\D Substitute the device-name (name) portion of the corresponding devinfo node-name.</p> <p>\An Substitute the <i>n</i>th component of the address component of the corresponding devinfo node name. Sub-components are separated by commas, and sub-component 0 is the whole address component.</p> <p>\Mn Substitute the <i>n</i>th sub-component of the minor component of the corresponding devinfo node name. Sub-components are separated by commas, and sub-component 0 is the whole minor component.</p> <p>\Nn Substitute the value of a 'counter' starting at <i>n</i>. There can be only one counter for each dev-spec, and counter-values will be selected so they are as low as possible while not colliding with already-existing link names.</p> <p>In a dev-spec the counter sequence should not be followed by a digit, either explicitly or as a result of another escape-sequence expansion. If this occurs, it would not be possible to correctly match already-existing links to their counter entries, since it would not be possible to unambiguously parse the already-existing /dev-name.</p>
<i>extra-dev-link</i>	<p>Optional specification of an extra /dev link that points to the initial /dev link (specified in field 2). This field may contain a counter escape-sequence (as described for the <i>dev-spec</i> field) but may not contain any of the other escape-sequences. It provides a way to specify an alias of a particular /dev name.</p>

OPTIONS	<p>-d Debugging mode – print out all devinfo nodes found, and indicate what links would be created, but do not do anything.</p> <p>-r rootdir Use <i>rootdir</i> as the root of the /dev and /devices directories under which the device nodes and links are created. Changing the root directory does not change the location of the /etc/devlink.tab default table, nor is the root directory applied to the filename supplied to the -t option.</p> <p>-t table-file Set the table file used by devlinks to specify the links that must be created. If this option is not given, /etc/devlink.tab is used. This option gives a way to instruct devlinks just to perform a particular piece of work, since just the links-types that devlinks is supposed to create can be specified in a command-file and fed to devlinks.</p>						
ERRORS	<p>If devlinks finds an error in a line of the <i>table-file</i> it prints a warning message on its standard output and goes on to the next line in the <i>table-file</i> without performing any of the actions specified by the erroneous rule.</p> <p>If it cannot create a link for some filesystem-related reason it prints an error-message and continues with the current rule.</p> <p>If it cannot read necessary data it prints an error message and continues with the next <i>table-file</i> line.</p>						
EXAMPLES	<p>Example /etc/devlink.tab fields are:</p> <pre> type=pseudo;name=win win\M0 type=ddi_display framebuffer/\M0 fb\N0 </pre> <p>The first example states that all devinfo nodes of type pseudo with a name component of win will be linked to /dev/winx, where <i>x</i> is the minor-component of the <i>devinfo-name</i> (this is always a single-digit number for the win driver).</p> <p>The second example states that all devinfo nodes of type ddi_display will be linked to entries under the /dev/framebuffer directory, with names identical to the entire minor component of the /devices name. In addition an extra link will be created pointing from /dev/fbn to the entry under /dev/framebuffer. This entry will use a counter to end the name.</p>						
FILES	<table border="0"> <tr> <td>/dev</td> <td>entries for the miscellaneous devices for general use</td> </tr> <tr> <td>/devices</td> <td>device nodes</td> </tr> <tr> <td>/etc/devlink.tab</td> <td>the default rule-file</td> </tr> </table>	/dev	entries for the miscellaneous devices for general use	/devices	device nodes	/etc/devlink.tab	the default rule-file
/dev	entries for the miscellaneous devices for general use						
/devices	device nodes						
/etc/devlink.tab	the default rule-file						
SEE ALSO	ddi_create_minor_node(9F) , disks(1M) , drvconfig(1M) , ports(1M) , tapes(1M)						
BUGS	It is very easy to construct mutually-contradictory link specifications, or specifications that can never be matched. The program does not check for these conditions.						

NAME	devnm – device name
SYNOPSIS	/usr/sbin/devnm <i>name</i> [<i>name...</i>]
AVAILABILITY	SUNWcsu
DESCRIPTION	The devnm command identifies the special file associated with the mounted file system where the argument <i>name</i> resides. One or more <i>name</i> can be specified.
EXAMPLE	The command: /usr/sbin/devnm /usr produces: /dev/dsk/c0t3d0s6 /usr if /usr is mounted on /dev/dsk/c0t3d0s6 .
FILES	/dev/dsk/* /etc/mnttab
SEE ALSO	mnttab(4)

NAME	devreserv – reserve devices for exclusive use
SYNOPSIS	devreserv [<i>key</i> [<i>devicelist</i> ...]]
DESCRIPTION	<p>devreserv reserves devices for exclusive use. When the device is no longer required, use devfree to release it.</p> <p>devreserv reserves at most one device per <i>devicelist</i>. Each list is searched in linear order until the first available device is found. If a device cannot be reserved from each list, the entire reservation fails.</p> <p>When devreserv is invoked without arguments, it lists the devices that are currently reserved and shows to which key it was reserved. When devreserv is invoked with only the <i>key</i> argument, it lists the devices that are currently reserved to that key.</p>
OPTIONS	<p><i>key</i> Designates a unique key on which the device will be reserved. The key must be a positive integer.</p> <p><i>devicelist</i> Defines a list of devices that devreserv will search to find an available device. (The list must be formatted as a single argument to the shell.)</p>
EXAMPLES	<p>To reserve a floppy disk and a cartridge tape:</p> <pre>\$ key=\$\$ \$ echo "The current Process ID is equal to: \$key" The Current Process ID is equal to: 10658 \$ devreserv \$key diskette1 ctape1</pre> <p>To list all devices currently reserved:</p> <pre>\$ devreserv disk1 2423 diskette1 10658 ctape1 10658</pre> <p>To list all devices currently reserved to a particular key:</p> <pre>\$ devreserv \$key diskette1 ctape1</pre>
ERRORS	<p>The command will exit with one of the following values:</p> <p>0 Successful completion of the task.</p> <p>1 Command syntax incorrect, invalid option used, or internal error occurred.</p> <p>2 Device table or device reservation table could not be opened for reading.</p> <p>3 Device reservation request could not be fulfilled.</p>

FILES /etc/device.tab
/etc/devlkfile

SEE ALSO devfree(1M)

NOTES The commands **devreserv** and **devfree** are used to manage the availability of devices on a system. Their use is on a participatory basis and they do not place any constraints on the actual access to the device. They serve as a centralized bookkeeping point for those who wish to use them. To summarize, devices which have been reserved cannot be used by processes which utilize the device reservation functions until the reservation has been canceled. However, processes that do not use device reservation may use a device that has been reserved since such a process would not have checked for its reservation status.

NAME	df – report number of free disk blocks and files
SYNOPSIS	<pre> /usr/bin/df [-F <i>FSType</i>] [-abegklntV] [-o <i>FSType-specific_options</i>] [<i>directory</i> <i>block_device</i> <i>resource</i> ...] /usr/xpg4/bin/df [-F <i>FSType</i>] [-abegklntV] [-o <i>FSType-specific_options</i>] [<i>directory</i> <i>block_device</i> <i>resource</i> ...] </pre>
AVAILABILITY	
/usr/bin/df	SUNWcsu
/usr/xpg4/bin/df	SUNWxcu4
DESCRIPTION	<p>The df command displays the amount of disk space occupied by mounted or unmounted file systems, directories, or mounted resources, the amount of used and available space, and how much of the file system's total capacity has been used.</p> <p><i>directory</i> represents a valid directory name. If <i>directory</i> is specified, df reports on the file system that contains <i>directory</i>. <i>block_device</i> represents a block special device (for example, /dev/dsk/c1d0s7); if <i>block_device</i> is specified, the corresponding file system need not be mounted. <i>resource</i> is an NFS resource name.</p> <p>Used without operands or options, df reports on all mounted file systems.</p>
OPTIONS	<p>The following options are supported:</p> <ul style="list-style-type: none"> -a Report on all filesystems including ones whose entries in /etc/mnttab (see mnttab(4)) have the ignore option set. -b Print the total number of kilobytes free. -e Print only the number of files free. -F <i>FSType</i> Specify the <i>FSType</i> on which to operate. This is only needed if the file system is unmounted. The <i>FSType</i> should be specified here or be determinable from /etc/vfstab (see vfstab(4)) have the by matching the <i>directory</i>, <i>block_device</i>, or <i>resource</i> with an entry in the table, or by consulting /etc/default/fs. See default_fs(4). -g Print the entire statvfs(2) structure. This option is used only for mounted file systems. It cannot be used with the -o option. This option will override the -b, -e, -k, -n, -P, and -t options. -k Print the allocation in kbytes. The output consists of one line of information for each specified file system. This information includes the file system name, the total space allocated in the file system, the amount of space allocated to existing files, the total amount of space available for the creation of new files by unprivileged users, and the percentage of normally available space that is currently allocated to all files on the file system. This option will override the -b, -e, -n, and -t options. -l Report on local file systems only. This option is used only for mounted

file systems. It cannot be used with the **-o** option.

- n** Print only the *FSType* name. Invoked with no operands, this option prints a list of mounted file system types. This option is used only for mounted file systems. It cannot be used with the **-o** option.
- o *FSType-specific_options*** Specify *FSType-specific* options. These options are comma-separated, with no intervening spaces. See the manual page for the *FSType-specific* command for details.
- P** Same as **-k** except in 512-byte units.
- t** Print full listings with totals. This option will override the **-b**, **-e**, and **-n** options.
- V** Echo the complete set of file system specific command lines, but do not execute them. The command line is generated by using the options and operands provided by the user and adding to them information derived from */etc/mnttab*, */etc/vfstab*, or */etc/default/fs*. This option may be used to verify and validate the command line.

OPERANDS

The following operands are supported:

- directory* represents a valid directory name. **df** reports on the file system that contains *directory*.
- block_device* represents a block special device (for example, */dev/dsk/c1d0s7*); the corresponding file system need not be mounted.
- resource* represents an NFS resource name.

EXAMPLES

The following example writes portable information about the */usr* file system:

```
example% /usr/xpg4/bin/df -P /usr
```

Assuming that */usr/src* is part of the */usr* file system, the following will do the same as the previous example:

```
example% /usr/xpg4/bin/df -P /usr/src
```

ENVIRONMENT

See **environ(5)** for descriptions of the following environment variables that affect the execution of **df**: **LC_CTYPE**, **LC_MESSAGES**, and **NLSPATH**.

EXIT STATUS

The following exit values are returned:

- 0** Successful completion.
- >0** An error occurred.

FILES

- /dev/dsk/** disk devices
- /etc/default/fs* default local file system type. Default values can be set for the following flags in */etc/default/fs*. For example: **LOCAL=ufs**

LOCAL: The default partition for a command if no *FSType* is specified.

/etc/mnttab mount table

/etc/vfstab list of default parameters for each file system

SEE ALSO **find(1)**, **mount(1M)**, **statvfs(2)**, **default_fs(4)**, **mnttab(4)**, **vfstab(4)**, **environ(5)**
Manual pages for the *FSType*-specific modules of **df**.

NOTES The **-F** option is intended for use with unmounted file systems.
This command may not be supported for all *FSTypes*.

NAME	df_ufs – report free disk space on ufs file systems
SYNOPSIS	df -F ufs [<i>generic_options</i>] [-o i] [<i>directory</i> <i>special</i>]
DESCRIPTION	<p>df displays the amount of disk space occupied by ufs file systems, the amount of used and available space, and how much of the file system's total capacity has been used.</p> <p>Note that the amount of space reported as used and available is less than the amount of space in the file system; this is because the system reserves a fraction of the space in the file system to allow its file system allocation routines to work well. The amount reserved is typically about 10%; this may be adjusted using tunefs(1M). When all the space on the file system except for this reserve is in use, only the super-user can allocate new files and data blocks to existing files. When the file system is overallocated in this way, df may report that the file system is more than 100% utilized.</p> <p>If neither <i>directory</i> nor <i>special</i> is specified, df displays information for all mounted ufs file systems.</p>
OPTIONS	<p><i>generic_options</i> Options supported by the generic df command. See df(1M) for a description of these options.</p> <p>-o Specify ufs file system specific options. The available option is:</p> <p style="padding-left: 2em;">i Report the number of used and free inodes. This option may not be used with <i>generic_options</i>.</p>
FILES	/etc/mnttab list of file systems currently mounted
SEE ALSO	df(1M) , tunefs(1M) , fs_ufs(4) , mnttab(4)
NOTES	df calculates its results differently for mounted and unmounted file systems. For unmounted systems, the numbers reflect the 10% reservation mentioned above; this reservation is not reflected in df output for mounted file systems. For this reason, the available space reported by the generic command may differ from the available space reported by this module.

NAME	dfmounts – display mounted resource information								
SYNOPSIS	dfmounts [-F <i>FSType</i>] [-h] [-o <i>specific_options</i>] [<i>restriction ...</i>]								
AVAILABILITY	SUNWcsu								
DESCRIPTION	<p>dfmounts shows the local resources shared through a distributed file system <i>FSType</i> along with a list of clients that have the resource mounted. If <i>restriction</i> is not specified, dfmounts shows file systems that are currently shared on any NFS server. <i>specific_options</i> as well as the availability and semantics of <i>restriction</i> are specific to particular distributed file system types.</p> <p>If dfmounts is entered without arguments, all remote resources currently mounted on the local system are displayed, regardless of file system type.</p>								
dfmounts Output	<p>The output of dfmounts consists of an optional header line (suppressed with the -h flag) followed by a list of lines containing whitespace-separated fields. For each resource, the fields are:</p> <p style="text-align: center;"><i>resource server pathname clients ...</i></p> <p>where:</p> <table border="0" style="margin-left: 40px;"> <tr> <td style="padding-right: 20px;"><i>resource</i></td> <td>Specifies the resource name that must be given to the mount(1M) command.</td> </tr> <tr> <td><i>server</i></td> <td>Specifies the system from which the resource was mounted.</td> </tr> <tr> <td><i>pathname</i></td> <td>Specifies the pathname that must be given to the share(1M) command.</td> </tr> <tr> <td><i>clients</i></td> <td>Is a comma-separated list of systems that have mounted the resource. Clients are listed in the form <i>domain.</i>, <i>domain.system</i>, or <i>system</i>, depending on the file system type.</td> </tr> </table> <p>A field may be null. Each null field is indicated by a hyphen (-) unless the remainder of the fields on the line are also null; in which case, the hyphen may be omitted.</p> <p>Fields with whitespace are enclosed in quotation marks (" ").</p>	<i>resource</i>	Specifies the resource name that must be given to the mount (1M) command.	<i>server</i>	Specifies the system from which the resource was mounted.	<i>pathname</i>	Specifies the pathname that must be given to the share (1M) command.	<i>clients</i>	Is a comma-separated list of systems that have mounted the resource. Clients are listed in the form <i>domain.</i> , <i>domain.system</i> , or <i>system</i> , depending on the file system type.
<i>resource</i>	Specifies the resource name that must be given to the mount (1M) command.								
<i>server</i>	Specifies the system from which the resource was mounted.								
<i>pathname</i>	Specifies the pathname that must be given to the share (1M) command.								
<i>clients</i>	Is a comma-separated list of systems that have mounted the resource. Clients are listed in the form <i>domain.</i> , <i>domain.system</i> , or <i>system</i> , depending on the file system type.								
OPTIONS	<p>-F <i>FSType</i> Specify filesystem type. Defaults to the first entry in <i>/etc/dfs/fstypes</i>.</p> <p style="margin-left: 100px;">Note: currently the only valid <i>FSType</i> is nfs.</p> <p>-h Suppress header line in output.</p> <p>-o <i>specific_options</i> Specify options specific to the filesystem provided by the -F option.</p> <p style="margin-left: 100px;">Note: currently no options are supported.</p>								

FILES /etc/dfs/fstypes file system types

SEE ALSO dfshares(1M), mount(1M), share(1M), unshare(1M)

NAME	dfmounts_nfs – display mounted NFS resource information								
SYNOPSIS	dfmounts [-F nfs] [-h] [<i>server ...</i>]								
DESCRIPTION	<p>dfmounts shows the local resources shared through NFS, along with the list of clients that have mounted the resource. The -F flag may be omitted if NFS is the only file system type listed in the file /etc/dfs/fstypes.</p> <p>dfmounts without options, displays all remote resources mounted on the local system, regardless of file system type.</p> <p>The output of dfmounts consists of an optional header line (suppressed with the -h flag) followed by a list of lines containing whitespace-separated fields. For each resource, the fields are:</p> <p style="text-align: center;"><i>resource server pathname clients ...</i></p> <p>where</p> <table border="0" style="margin-left: 40px;"> <tr> <td><i>resource</i></td> <td>Does not apply to NFS. Printed as a hyphen (-).</td> </tr> <tr> <td><i>server</i></td> <td>Specifies the system from which the resource was mounted.</td> </tr> <tr> <td><i>pathname</i></td> <td>Specifies the pathname that must be given to the share(1M) command.</td> </tr> <tr> <td><i>clients</i></td> <td>Is a comma-separated list of systems that have mounted the resource.</td> </tr> </table>	<i>resource</i>	Does not apply to NFS. Printed as a hyphen (-).	<i>server</i>	Specifies the system from which the resource was mounted.	<i>pathname</i>	Specifies the pathname that must be given to the share(1M) command.	<i>clients</i>	Is a comma-separated list of systems that have mounted the resource.
<i>resource</i>	Does not apply to NFS. Printed as a hyphen (-).								
<i>server</i>	Specifies the system from which the resource was mounted.								
<i>pathname</i>	Specifies the pathname that must be given to the share(1M) command.								
<i>clients</i>	Is a comma-separated list of systems that have mounted the resource.								
OPTIONS	<p>-F nfs Specifies the nfs-FSType.</p> <p>-h Suppress header line in output.</p> <p><i>server</i> Displays information about the resources mounted from each server, where <i>server</i> can be any system on the network. If no <i>server</i> is specified, the <i>server</i> is assumed to be the local system.</p>								
FILES	/etc/dfs/fstypes								
SEE ALSO	mount(1M) , share(1M) , unshare(1M)								

NAME	dfshares – list available resources from remote or local systems								
SYNOPSIS	dfshares [-F <i>FSType</i>] [-h] [-o <i>specific_options</i>] [<i>server</i> ...]								
AVAILABILITY	SUNWcsu								
DESCRIPTION	<p>dfshares provides information about resources available to the host through a distributed file system of type <i>FSType</i>. <i>specific_options</i> as well as the semantics of <i>server</i> are specific to particular distributed file systems.</p> <p>If dfshares is entered without arguments, all resources currently shared on the local system are displayed, regardless of file system type.</p> <p>The output of dfshares consists of an optional header line (suppressed with the -h flag) followed by a list of lines containing whitespace-separated fields. For each resource, the fields are:</p> <p style="text-align: center;"><i>resource server access transport</i></p> <p>where</p> <table border="0" style="margin-left: 2em;"> <tr> <td style="padding-right: 1em;"><i>resource</i></td> <td>Specifies the resource name that must be given to the mount(1M) command.</td> </tr> <tr> <td style="padding-right: 1em;"><i>server</i></td> <td>Specifies the name of the system that is making the resource available.</td> </tr> <tr> <td style="padding-right: 1em;"><i>access</i></td> <td>Specifies the access permissions granted to the client systems, either ro (for read-only) or rw (for read/write). If dfshares cannot determine access permissions, a hyphen (-) is displayed.</td> </tr> <tr> <td style="padding-right: 1em;"><i>transport</i></td> <td>Specifies the transport provider over which the resource is shared.</td> </tr> </table> <p>A field may be null. Each null field is indicated by a hyphen (-) unless the remainder of the fields on the line are also null; in which case, the hyphen may be omitted.</p>	<i>resource</i>	Specifies the resource name that must be given to the mount (1M) command.	<i>server</i>	Specifies the name of the system that is making the resource available.	<i>access</i>	Specifies the access permissions granted to the client systems, either ro (for read-only) or rw (for read/write). If dfshares cannot determine access permissions, a hyphen (-) is displayed.	<i>transport</i>	Specifies the transport provider over which the resource is shared.
<i>resource</i>	Specifies the resource name that must be given to the mount (1M) command.								
<i>server</i>	Specifies the name of the system that is making the resource available.								
<i>access</i>	Specifies the access permissions granted to the client systems, either ro (for read-only) or rw (for read/write). If dfshares cannot determine access permissions, a hyphen (-) is displayed.								
<i>transport</i>	Specifies the transport provider over which the resource is shared.								
OPTIONS	<p>-F <i>FSType</i> Specify filesystem type. Defaults to the first entry in <i>/etc/dfs/fstypes</i>.</p> <p>-h Suppress header line in output.</p> <p>-o <i>specific_options</i> Specify options specific to the filesystem provided by the -F option.</p>								
FILES	<i>/etc/dfs/fstypes</i>								
SEE ALSO	dfmounts (1M), mount (1M), share (1M), unshare (1M)								

NAME	dfshares_nfs – list available NFS resources from remote systems								
SYNOPSIS	dfshares [-F nfs] [-h] [<i>server ...</i>]								
DESCRIPTION	<p>dfshares provides information about resources available to the host through NFS. The -F flag may be omitted if NFS is the first file system type listed in the file /etc/dfs/fstypes.</p> <p>The query may be restricted to the output of resources available from one or more servers.</p> <p>dfshares without arguments displays all resources shared on the local system, regardless of file system type.</p> <p>Specifying <i>server</i> displays information about the resources shared by each server. <i>Server</i> can be any system on the network. If no server is specified, then <i>server</i> is assumed to be the local system.</p> <p>The output of dfshares consists of an optional header line (suppressed with the -h flag) followed by a list of lines containing whitespace-separated fields. For each resource, the fields are:</p> <p style="text-align: center;"><i>resource server access transport</i></p> <p>where</p> <table border="0" style="margin-left: 40px;"> <tr> <td style="padding-right: 20px;"><i>resource</i></td> <td>Specifies the resource name that must be given to the mount(1M) command.</td> </tr> <tr> <td><i>server</i></td> <td>Specifies the system that is making the resource available.</td> </tr> <tr> <td><i>access</i></td> <td>Specifies the access permissions granted to the client systems; however, dfshares cannot determine this information for an NFS resource and populates the field with a hyphen (-).</td> </tr> <tr> <td><i>transport</i></td> <td>Specifies the transport provider over which the <i>resource</i> is shared; however, dfshares cannot determine this information for an NFS resource and populates the field with a hyphen (-).</td> </tr> </table> <p>A field may be null. Each null field is indicated by a hyphen (-) unless the remainder of the fields on the line are also null; in which case, the hyphen may be omitted.</p>	<i>resource</i>	Specifies the resource name that must be given to the mount(1M) command.	<i>server</i>	Specifies the system that is making the resource available.	<i>access</i>	Specifies the access permissions granted to the client systems; however, dfshares cannot determine this information for an NFS resource and populates the field with a hyphen (-).	<i>transport</i>	Specifies the transport provider over which the <i>resource</i> is shared; however, dfshares cannot determine this information for an NFS resource and populates the field with a hyphen (-).
<i>resource</i>	Specifies the resource name that must be given to the mount(1M) command.								
<i>server</i>	Specifies the system that is making the resource available.								
<i>access</i>	Specifies the access permissions granted to the client systems; however, dfshares cannot determine this information for an NFS resource and populates the field with a hyphen (-).								
<i>transport</i>	Specifies the transport provider over which the <i>resource</i> is shared; however, dfshares cannot determine this information for an NFS resource and populates the field with a hyphen (-).								
OPTIONS	<p>-F nfs Specify the NFS file system type</p> <p>-h Suppress header line in output.</p>								
FILES	/etc/dfs/fstypes								
SEE ALSO	mount(1M) , share(1M) , unshare(1M)								

NAME	discover4x – analyze the SunOS 4.x client support after converting to Solaris 2.x		
SYNOPSIS	<code>/usr/sbin/discover4x [export_root]</code>		
AVAILABILITY	SUNWhinst		
DESCRIPTION	<p>discover4x analyzes the support that remains for SunOS 4.x clients after the server has been converted to Solaris 2.x.</p> <p>The discover4x command is most commonly used after a SunOS 4.x server has been upgraded to a Solaris 2.x server. This command looks at the SunOS 4.x client support that exists and creates the databases that are required for installation of SunOS 4.x diskless/dataless clients on this Solaris 2.x server. If client support for a given architecture is missing, discover4x attempts to notify the user that they will have to re-install this support using install4x(1M). If you are converting clients that have the same architecture as the server that was converted to Solaris 2.x, you must re-install that architecture using the install4x(1M) command.</p>		
OPTIONS	<table><tr><td><i>export_root</i></td><td>Specify the location of the base directory of the client hierarchy that discover4x should analyze. If unspecified, the default is /export.</td></tr></table>	<i>export_root</i>	Specify the location of the base directory of the client hierarchy that discover4x should analyze. If unspecified, the default is /export .
<i>export_root</i>	Specify the location of the base directory of the client hierarchy that discover4x should analyze. If unspecified, the default is /export .		
SEE ALSO	<p>convert4x(1M), install4x(1M)</p> <p><i>SPARC: Installing Solaris Software</i></p> <p><i>x86: Installing Solaris Software</i></p>		

NAME	disks – adds /dev entries for hard disks attached to the system
SYNOPSIS	<code>/usr/sbin/disks [-r <i>rootdir</i>]</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>disks creates symbolic links in the <code>/dev/dsk</code> and <code>/dev/rdsk</code> directories pointing to the actual disk device special files under the <code>/devices</code> directory tree. It performs the following steps:</p> <ol style="list-style-type: none"> 1. disks searches the kernel device tree to see what hard disks are attached to the system. It notes the <code>/devices</code> pathnames for the partitions on the drive and determines the physical component of the corresponding <code>/dev/dsk</code> or <code>/dev/rdsk</code> name. 2. The <code>/dev/dsk</code> and <code>/dev/rdsk</code> directories are checked for disk partition entries – that is, symbolic links with names of the form <code>cN[tN]dNsN</code>, where <i>N</i> represents a decimal number. <i>cN</i> is the logical controller number, an arbitrary number assigned by this program to designate a particular disk controller. The first controller found on the first occasion this program is run on a system, is assigned number 0. <i>tN</i> is the bus-address number of a subsidiary controller attached to a peripheral bus such as SCSI or IPI (the target number for SCSI, and the facility number for IPI controllers). <i>dN</i> is the number of the disk attached to the controller, and <i>sN</i> is the partition, or <i>slice</i>, number of the entry. 3. If only some of the disk partition entries are found in <code>/dev/dsk</code> for a disk that has been found under the <code>/devices</code> directory tree, disks creates the missing symbolic links. If none of the partition entries for a particular disk are found in <code>/dev/dsk</code>, disks checks to see if any entries exist for other disks attached to the same controller, and if so, creates new entries using the same controller number as used for other disks on the same controller. If no other <code>/dev/dsk</code> entries are found for partitions of disks belonging to the same physical controller as the current disk, disks assigns the lowest-unused controller number and creates entries for the disk partitions using this newly-assigned controller number. <p>disks is called each time the system is reconfiguration-booted. disks can only be run after <code>drvconfig(1M)</code> is run, since <code>drvconfig(1M)</code> builds the kernel data structures and the <code>/devices</code> tree.</p> <p>Of interest to device driver writers are the devices that disks considers disk devices. Device nodes of type <code>DDI_NT_BLOCK</code>, <code>DDI_NT_BLOCK_CHAN</code>, <code>DDI_NT_CD</code>, and <code>DDI_NT_CD_CHAN</code> are considered disk devices (see <code>ddi_create_minor_node(9F)</code> for more information on node types).</p>
OPTIONS	<p><code>-r <i>rootdir</i></code> Cause disks to presume that the <code>/dev/dsk</code>, <code>/dev/rdsk</code> and <code>/devices</code> directory trees are found under <i>rootdir</i>, not directly under <code>/</code>.</p>

ERRORS If **disks** finds entries of a particular logical controller linked to different physical controllers, it prints an error message and exits without making any changes to the **/dev** directory, since it cannot determine which of the two alternative logical-to-physical mappings is correct. The links should be manually corrected or removed before another reconfiguration-boot is performed.

FILES **/dev/dsk/*** entries for the hard disk for general use
/dev/rdisk/*
/devices/* device nodes

SEE ALSO **devlinks(1M)**, **drvconfig(1M)**, **ports(1M)**, **tapes(1M)**, **ddi_create_minor_node(9F)**

NAME	diskscan – perform surface analysis
SYNOPSIS	diskscan [-W] [-n] [-y] <i>raw_device</i>
AVAILABILITY	x86 SUNWcsu
DESCRIPTION	<p>diskscan is used by the system administrator to perform surface analysis on a portion of a hard disk. The disk portion may be a raw partition or slice; it is identified using its raw device name. By default, the specified portion of the disk is read (non-destructive) and errors reported on standard error. In addition, a progress report is printed on standard out. The list of bad blocks should be saved in a file and later fed into addbadsec(1M), which will remap them.</p> <p>-W Causes diskscan to perform write and read surface analysis. This type of surface analysis is destructive and should be invoked with caution.</p> <p>-n Causes diskscan to suppress linefeeds when printing progress information on standard out.</p> <p>-y Causes diskscan to suppress the warning regarding destruction of existing data that is issued when -W is used.</p>
FILES	The raw device should be /dev/rdisk/c?t?d?[ps]? .
SEE ALSO	fdisk (1M), fmthard (1M), addbadsec (1M)

NAME	dispadmin – process scheduler administration
SYNOPSIS	<p>dispadmin -l dispadmin -c class -g [-r res] dispadmin -c class -s file</p>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>The dispadmin command displays or changes process scheduler parameters while the system is running.</p> <p>dispadmin does limited checking on the values supplied in <i>file</i> to verify that they are within their required bounds. The checking, however, does not attempt to analyze the effect that the new values have on the performance of the system. Inappropriate values can have a negative effect on system performance. (See <i>System Administration Guide, Volume II</i>.)</p>
OPTIONS	<p>-l Lists the scheduler classes currently configured in the system.</p> <p>-c class Specifies the class whose parameters are to be displayed or changed. Valid <i>class</i> values are: RT for the real-time class, TS for the time-sharing class, and IA for the inter-active class. The time-sharing and inter-active classes share the same scheduler, so changes to the scheduling parameters of one will change those of the other.</p> <p>-g Gets the parameters for the specified class and writes them to the standard output. Parameters for the real-time class are described in rt_dptbl(4). Parameters for the time-sharing and inter-active classes are described in ts_dptbl(4).</p> <p>-r res When using the -g option you may also use the -r option to specify a resolution to be used for outputting the time quantum values. If no resolution is specified, time quantum values are in milliseconds. If <i>res</i> is specified it must be a positive integer between 1 and 100000000 inclusive, and the resolution used is the reciprocal of <i>res</i> in seconds. For example, a <i>res</i> value of 10 yields time quantum values expressed in tenths of a second; a <i>res</i> value of 1000000 yields time quantum values expressed in microseconds. If the time quantum cannot be expressed as an integer in the specified resolution, it is rounded up to the next integral multiple of the specified resolution.</p> <p>-s file Sets scheduler parameters for the specified class using the values in <i>file</i>. These values overwrite the current values in memory—they become the parameters that control scheduling of processes in the specified class. The values in <i>file</i> must be in the format output by the -g option. Moreover, the values must describe a table that is the same size (has same number of priority levels) as the table being overwritten.</p>

Super-user privileges are required in order to use the `-s` option.

Note: The `-g` and `-s` options are mutually exclusive: you may not retrieve the table at the same time you are overwriting it.

EXAMPLES

The following command retrieves the current scheduler parameters for the real-time class from kernel memory and writes them to the standard output. Time quantum values are in microseconds.

```
dispadmin -c RT -g -r 1000000
```

The following command overwrites the current scheduler parameters for the real-time class with the values specified in `rt.config`.

```
dispadmin -c RT -s rt.config
```

The following command retrieves the current scheduler parameters for the time-sharing class from kernel memory and writes them to the standard output. Time quantum values are in nanoseconds.

```
dispadmin -c TS -g -r 1000000000
```

The following command overwrites the current scheduler parameters for the time-sharing class with the values specified in `ts.config`.

```
dispadmin -c TS -s ts.config
```

SEE ALSO

`priocntl(1)`, `priocntl(2)`, `rt_dptbl(4)`, `ts_dptbl(4)`,

System Interfaces Guide

DIAGNOSTICS

`dispadmin` prints an appropriate diagnostic message if it fails to overwrite the current scheduler parameters due to lack of required permissions or a problem with the specified input file.

NAME	dmesg – collect system diagnostic messages to form error log
SYNOPSIS	<code>/usr/bin/dmesg [-]</code> <code>/usr/sbin/dmesg [-]</code>
AVAILABILITY	SUNWesu
DESCRIPTION	Note: dmesg is made obsolete by syslogd(1M) for maintenance of the system error log. dmesg looks in a system buffer for recently printed diagnostic messages and prints them on the standard output. The messages are those printed or logged by the system when errors occur. If the <code>'-'</code> flag is given, then dmesg computes (incrementally) the new messages since the last time it was run and places these on the standard output.
FILES	<code>/var/adm/msgbuf</code> scratch file for memory of <code>'-'</code> option
SEE ALSO	syslogd(1M)

NAME	dminfo – report information about a device entry in a device maps file
SYNOPSIS	<p>dminfo [-v] [-a] [-f <i>pathname</i>]</p> <p>dminfo [-v] [-a] [-f <i>pathname</i>] -n <i>dev-name</i> ...</p> <p>dminfo [-v] [-a] [-f <i>pathname</i>] -d <i>dev-path</i> ...</p> <p>dminfo [-v] [-a] [-f <i>pathname</i>] -t <i>dev-type</i> ...</p> <p>dminfo [-v] [-f <i>pathname</i>] -u <i>dm-entry</i></p>
AVAILABILITY	The functionality described in this man page is available only if the Basic Security Module (BSM) has been enabled. See bsmconv(1M) for more information.
DESCRIPTION	dminfo reports and updates information about the device_maps(4) file.
OPTIONS	<p>-v Verbose. Print the requested entry or entries, one line per entry, on the standard output. If no entries are specified, all are printed.</p> <p>-a Succeed if any of the requested entries are found. If used with -v, all entries that match the requested case(s) are printed.</p> <p>-f <i>pathname</i> Use a device_maps file with <i>pathname</i> instead of /etc/security/device_maps.</p> <p>-n <i>dev-name</i> Search by <i>dev-name</i>. Search device_maps(4) for a <i>device_name</i> field matching <i>dev-name</i>. This option cannot be used with -d, -t or -u.</p> <p>-d <i>dev-path</i> Search by <i>dev-path</i>. Search device_maps(4) for a device special path-name in the <i>device_list</i> field matching the <i>dev-path</i> argument. This option cannot be used with -n, -t or -u.</p> <p>-t <i>dev-type</i> Search by <i>dev-type</i>. Search device_maps(4) for a <i>device_type</i> field matching the given <i>dev-type</i>. This option cannot be used with -d, -n or -u.</p> <p>-u <i>dm-entry</i> Update the device_maps(4) file. This option is provided to add entries to the device_maps(4) file. The <i>dm-entry</i> must be a complete device_maps(4) file entry. The <i>dm-entry</i> has fields, as in the device_maps file. It uses the colon (:) as a field separator, and white space as the <i>device_list</i> subfield separators. The <i>dm-entry</i> is not made if any fields are missing, or if the <i>dm-entry</i> would be a duplicate. The default device maps file can be updated only by the super user.</p>
DIAGNOSTICS	dminfo returns an exit code of 0 if successful, 1 if the request failed, and 2 if the invocation syntax was incorrect.

FILES /etc/security/device_maps

SEE ALSO bsmconv(1M), device_maps(4)

NAME	domainname – set or display name of the current domain
SYNOPSIS	domainname [<i>name-of-domain</i>]
DESCRIPTION	<p>Without an argument, domainname displays the name of the current domain, which typically encompasses a group of hosts or passwd entries under the same administration.</p> <p>The domainname command is used by various components of Solaris to resolve names for types such as passwd, hosts and aliases. By default, various naming services such as NIS, NIS+, the Internet Domain Name Service (DNS) and sendmail(1M) use this domainname to resolve names. The domainname is normally a valid Internet domain name.</p> <p>The domainname for various naming services can also be set by other means. For example, ypinit can be used to specify a different domain for all NIS calls. The file /etc/resolv.conf can be used to specify a different domain for DNS lookups. For sendmail, the domainname can be specified through the <i>sendmail_vars</i> entry in the /etc/nsswitch.conf file, or through the /etc/mail/sendmail.cf file.</p> <p>Only the super-user can set the name of the domain by specifying the new domainname as an argument. The domain name of the machine is usually set during boot-time through the domainname command in the /etc/init.d/inetinit file. If the new domain name is not saved in the /etc/defaultdomain file, the machine will revert back to the old domain after rebooting.</p>
FILES	/etc/defaultdomain /etc/init.d/inetinit /etc/mail/sendmail.cf /etc/nsswitch.conf /etc/resolv.conf
SEE ALSO	nis+(1) , nischown(1) , nisspasswd(1) , hostconfig(1M) , named(1M) , nisaddcred(1M) , sendmail(1M) , ypinit(1M) , sys-unconfig(1M) , aliases(4) , hosts(4) , nsswitch.conf(4) , passwd(4)

NAME	drvconfig – configure the /devices directory
SYNOPSIS	drvconfig [-b] [-d] [-a <i>alias_name</i>] [-c <i>class_name</i>] [-i <i>drivername</i>] [-m <i>major_num</i>] [-n] [-r <i>rootdir</i>]
AVAILABILITY	SUNWcsu
DESCRIPTION	The default operation of drvconfig is to create the /devices directory tree that describes, in the filesystem namespace, the hardware layout of a particular machine. Hardware devices present on the machine and powered on as well as pseudo-drivers are represented under /devices . Normally this command is run automatically after a new driver has been installed (with add_drv (1M)) and the system has been rebooted.
/etc/minor_perm file	drvconfig reads the /etc/minor_perm file to obtain permission information and applies the permissions only to nodes which it has just created. It does not change permissions on already existing nodes. The format of the /etc/minor_perm file is as follows: <pre> <i>name</i>:<i>minor_name</i> <i>permissions</i> <i>owner</i> <i>group</i> </pre> <i>minor_name</i> may be the actual name of the minor node, or have shell metacharacters to represent several minor nodes (see sh (1)). For example: <pre> sd:* 0640 root sys zs: [a-z],cu 0600 uucp uucp mm:kmem 0640 root bin </pre> The above example sets all devices exported by the sd node to 0640 permissions, owned by root , with group sys . In the second example, devices such as a,cu and z,cu exported by the zs driver are set to 0600 permission, owned by uucp , with group uucp . The kmem device exported by the mm driver is set to 0640 permission, owned by root , with group bin .
OPTIONS	The following options may be of use to system administrators and driver developers: -d Debug. Print out debugging information about the devices it has found. -i <i>drivername</i> Only configure the devices for the named driver. The following options are used by the implementation of add_drv (1M) and rem_drv (1M), and may not be supported in future versions of Solaris: -b Add a new major number to name binding into the kernel's internal name_to_major tables. This option is not normally used directly, but is used by other utilities such as add_drv (1M). Use of the -b option requires that -i and -m be used also. No /devices entries are created. -n Do not try to load and attach any drivers, or if the -i option is given, do not try to attach the driver named <i>drivername</i> . -a <i>alias_name</i> Add the name <i>alias_name</i> to the list of aliases that this driver is known by. This option, if used, must be used with the -m <i>major_num</i> , the -b

and the **-i** *drivename* options.

- c** *class_name* The driver being added to the system exports the class *class_name*. This option is not normally used directly, but is used by other utilities. It is only effective when used with the **-b** option.
- m** *major_num* Specify the major number *major_num* for this driver to add to the kernel's **name_to_major** binding tables.
- p** Specify a path-to-instance file as an alternative to the default of **/etc/path_to_inst**.
- r** *rootdir* Build the device tree under the directory specified by *rootdir* instead of the default **/devices** directory.

FILES	/devices	device nodes directory
	/etc/minor_perm	minor mode permissions
	/etc/name_to_major	major number binding
	/etc/driver_classes	driver class binding file

SEE ALSO **sh(1)**, **add_drv(1M)**, **devlinks(1M)**, **disks(1M)**, **modinfo(1M)**, **modload(1M)**, **modunload(1M)**, **ports(1M)**, **rem_drv(1M)**, **tapes(1M)**, **path_to_inst(4)**

NOTES This document does not constitute an API. **/etc/minor_perm**, **/etc/name_to_major**, **/etc/driver_classes**, and **/devices** may not exist or may have different contents or interpretations in a future release. The existence of this notice does not imply that any other documentation that lacks this notice constitutes an API.

NAME	du – summarize disk usage
SYNOPSIS	/usr/bin/du [-adkr] [-s -o] [<i>file ...</i>] /usr/xpg4/bin/du [-a -s] [-krx] [<i>file ...</i>]
AVAILABILITY	
/usr/bin/du	SUNWcsu
/usr/xpg4/bin/du	SUNWxcu4
DESCRIPTION	<p>The du utility writes to standard output the size of the file space allocated to, and the size of the file space allocated to each subdirectory of, the file hierarchy rooted in each of the specified files. The size of the file space allocated to a file of type directory is defined as the sum total of space allocated to all files in the file hierarchy rooted in the directory plus the space allocated to the directory itself.</p> <p>Files with multiple links will be counted and written for only one entry. The directory entry that is selected in the report is unspecified. By default, file sizes are written in 512-byte units, rounded up to the next 512-byte unit.</p>
/usr/xpg4/bin/du	When du cannot obtain file attributes or read directories (see stat(2)), it will report an error condition and the final exit status will be affected.
OPTIONS	<p>The following options are supported by /usr/bin/du and /usr/xpg4/bin/du:</p> <ul style="list-style-type: none"> -a In addition to the default output, report the size of each file not of type directory in the file hierarchy rooted in the specified file. Regardless of the presence of the -a option, non-directories given as <i>file</i> operands will always be listed. -k Write the files sizes in units of 1024 bytes, rather than the default 512-byte units. -s Instead of the default output, report only the total sum for each of the specified files.
/usr/bin/du	<p>The following options are supported by /usr/bin/du only:</p> <ul style="list-style-type: none"> -d Do not cross filesystem boundaries. "du -d /" would report usage only on the root partition, for example. -o Do not add child directories' usage to a parent's total. Without this option, the usage listed for a particular directory is the space taken by the files in that directory, as well as the files in all directories beneath it. This option does nothing if -s is used. -r Generate messages about directories that cannot be read, files that cannot be opened, and so forth, rather than being silent (the default).
/usr/xpg4/bin/du	<p>The following options are supported by /usr/xpg4/bin/du only:</p> <ul style="list-style-type: none"> -r By default, generate messages about directories that cannot be read, files that cannot be opened, and so forth.

-x When evaluating file sizes, evaluate only those files that have the same device as the file specified by the *file* operand.

OPERANDS The following operand is supported:

file The path name of a file whose size is to be written. If no *file* is specified, the current directory is used.

OUTPUT The output from **du** consists of the amount of the space allocated to a file and the name of the file.

ENVIRONMENT See **environ(5)** for descriptions of the following environment variables that affect the execution of **du**: **LC_CTYPE**, **LC_MESSAGES**, and **NLSPATH**.

EXIT STATUS The following exit values are returned:

0 Successful completion.
>0 An error occurred.

SEE ALSO **ls(1)**, **stat(2)**, **environ(5)**

System Administration Guide, Volume II

NOTES A file with two or more links is counted only once. If, however, there are links between files in different directories where the directories are on separate branches of the file system hierarchy, **du** will count the excess files more than once.
Files containing holes will result in an incorrect block count.

NAME	edquota – edit user quotas for ufs file system				
SYNOPSIS	edquota [-p <i>proto_user</i>] <i>username</i> ... edquota -t				
DESCRIPTION	<p>edquota is a quota editor. One or more users may be specified on the command line. For each user a temporary file is created with an ASCII representation of the current disk quotas for that user for each mounted ufs file system that has a quotas file, and an editor is then invoked on the file. The quotas may then be modified, new quotas added, etc. Upon leaving the editor, edquota reads the temporary file and modifies the binary quota files to reflect the changes made.</p> <p>The editor invoked is vi(1) unless the EDITOR environment variable specifies otherwise. Only the super-user may edit quotas. In order for quotas to be established on a file system, the root directory of the file system must contain a file, owned by root, called quotas. (See quotaon(1M).)</p> <p><i>proto_user</i> and <i>username</i> can be numeric, corresponding to the uid of a user. Unassigned uids may be specified; unassigned names may not. In this way, default quotas can be established for users who are later assigned a uid.</p> <p>If no options are specified, the temporary file created will have one or more lines of the form</p> <pre style="margin-left: 40px;">fs <i>mount_point</i> blocks (soft =<i>number</i>, hard =<i>number</i>) inodes (soft =<i>number</i>, hard =<i>number</i>)</pre> <p>Where a block is considered to be a 1024 byte (1K) block.</p> <p>The <i>number</i> fields may be modified to reflect desired values.</p>				
OPTIONS	<p>-p Duplicate the quotas of the <i>proto_user</i> specified for each <i>username</i> specified. This is the normal mechanism used to initialize quotas for groups of users.</p> <p>-t Edit the soft time limits for each file system. If the time limits are zero, the default time limits in /usr/include/sys/fs/ufs_quota.h are used. The temporary file created will have one or more lines of the form</p> <pre style="margin-left: 40px;">fs <i>mount_point</i> blocks time limit = <i>number</i> <i>tmunit</i>, files time limit = <i>number</i> <i>tmunit</i></pre> <p><i>tmunit</i> may be one of “month”, “week”, “day”, “hour”, “min” or “sec”; characters appended to these keywords are ignored, so you may write “months” or “minutes” if you prefer. The <i>number</i> and <i>tmunit</i> fields may be modified to set desired values. Time limits are printed in the greatest possible time unit such that the value is greater than or equal to one. If “default” is printed after the <i>tmunit</i>, this indicates that the value shown is zero (the default).</p>				
FILES	<table border="0" style="width: 100%;"> <tr> <td style="padding-right: 20px;">quotas</td> <td>quota file at the file system root</td> </tr> <tr> <td>/etc/mnttab</td> <td>table of mounted file systems</td> </tr> </table>	quotas	quota file at the file system root	/etc/mnttab	table of mounted file systems
quotas	quota file at the file system root				
/etc/mnttab	table of mounted file systems				

SEE ALSO

vi(1), **quota(1M)**, **quotacheck(1M)**, **quotaon(1M)**, **repquota(1M)**, **quotactl(7I)**

NOTES

quotacheck(1M) *must* be invoked when setting initial quota limits for users; if not, the quota limit remains 0 and no changes made with **edquota** will take effect.

NAME	eeprom – EEPROM display and load utility
SYNOPSIS	<code>/usr/platform/platform-name/sbin/eeprom [-] [-f device] [parameter [=value] ...]</code>
AVAILABILITY	SPARC SUNWcsu eeprom is only available on OpenBoot-compatible SPARC systems
DESCRIPTION	eeprom displays or changes the values of parameters in the EEPROM. It processes parameters in the order given. When processing a <i>parameter</i> accompanied by a <i>value</i> , eeprom makes the indicated alteration to the EEPROM; otherwise it displays the <i>parameter</i> 's value. When given no parameter specifiers, eeprom displays the values of all EEPROM parameters. A '-' (hyphen) flag specifies that parameters and values are to be read from the standard input (one <i>parameter</i> or <i>parameter=value</i> per line). Only the super-user may alter the EEPROM contents. eeprom verifies the EEPROM checksums and complains if they are incorrect. <i>platform-name</i> is the name of the platform implementation and can be found using the -i option of uname(1) .
OPTIONS	<code>-f device</code> Use <i>device</i> as the EEPROM device.
NVRAM CONFIGURATION PARAMETERS	Not all OpenBoot systems support all parameters. Defaults may vary depending on the system and the PROM revision. auto-boot? If true , boot automatically after power-on or reset. Defaults to true . ansi-terminal? Configuration variable used to control the behavior of the terminal emulator. The value false makes the terminal emulator stop interpreting ANSI escape sequences, instead just echoing them to the output device. Default value: true . boot-command Command executed if auto-boot? is true . Default value is boot . boot-device Device from which to boot. Defaults to disk net . boot-file File to boot (an empty string lets the secondary booter choose default). Defaults to empty string. boot-from Boot device and file (OpenBoot PROM version 1.x only). Defaults to vmunix . boot-from-diag Diagnostic boot device and file (OpenBoot PROM version 1.x only). Defaults to le(unix) . diag-device Diagnostic boot source device. Defaults to net . diag-file File from which to boot in diagnostic mode. Defaults to empty string.

diag-level	Diagnostics level. Values include off , min , max and menus . There may be additional platform-specific values. When set to off , POST is not called. If POST is called, the value is made available as an argument to, and is interpreted by POST. The default value is platform-dependent .
diag-switch?	If true , run in diagnostic mode. Defaults to true .
fcode-debug?	If true , include name parameter for plug-in device FCodes. Defaults to false .
hardware-revision	System version information.
input-device	Power-on input device (usually keyboard , ttya , or ttyb). Defaults to keyboard .
keyboard-click?	If true enable keyboard click. Defaults to false .
keymap	Keymap for custom keyboard.
last-hardware-update	System update information.
load-base	Default load address for client programs. Default value is 16384 .
local-mac-address?	If true, network drivers use their own MAC address, not system's. Defaults to false .
mfg-mode	Manufacturing mode argument for POST. Possible values include off or chamber . The value is passed as an argument to POST. Default value: off .
mfg-switch?	If true, repeat system self-tests until interrupted with STOP-A. Defaults to false .
nvrामrc	Contents of NVRAMRC. Defaults to empty.
oem-banner	Custom OEM banner (enabled by setting oem-banner? to true). Defaults to empty string.
oem-banner?	If true , use custom OEM banner. Defaults to false .
oem-logo	Byte array custom OEM logo (enabled by setting oem-logo? to true). Displayed in hexadecimal.
oem-logo?	If true , use custom OEM logo (else, use Sun logo). Defaults to false .
output-device	Power-on input device (usually screen , ttya , or ttyb). Defaults to screen .
sbus-probe-list	Which SBus slots are probed and in what order. Defaults to 0123 .
screen-#columns	Number of on-screen columns (characters/line). Defaults to 80 .
screen-#rows	Number of on-screen rows (lines). Defaults to 34 .
scsi-initiator-id	SCSI bus address of host adapter, range 0-7. Defaults to 7.

sd-targets	Map SCSI disk units (OpenBoot PROM version 1.x only). Defaults to 31204567 , which means that unit 0 maps to target 3 , unit 1 maps to target 1 , and so on.
security-#badlogins	Number of incorrect security password attempts.
security-mode	Firmware security level (options: none , command , or full). If set to command or full , system will prompt for PROM security password. Defaults to none .
security-password	Firmware security password (never displayed). Can be set only when security-mode is set to command or full . example# eeprom security-password= Changing PROM password: New password: Retype new password:
selftest-#megs	Metabytes of RAM to test. Ignored if diag-switch? is true . Defaults to 1 .
skip-vme-loopback?	If true , POST does not do VMEbus loopback tests. Defaults to false .
st-targets	Map SCSI tape units (OpenBoot PROM version 1.x only). Defaults to 45670123 , which means that unit 0 maps to target 4 , unit 1 maps to target 5 , and so on.
sunmon-compat?	If true , display Restricted Monitor prompt (>). Defaults to false .
testarea	One-byte scratch field, available for read/write test. Defaults to 0 .
tpe-link-test?	Enable 10baseT link test for built-in twisted pair Ethernet. Defaults to true .
ttya-mode	TTYA (baud rate, #bits, parity, #stop, handshake). Defaults to 9600,8,n,1,- . Fields, in left-to-right order, are: baud rate: 110, 300, 1200, 4800, 9600 ... data bits: 5, 6, 7, 8 parity: n(none), e(even), o(odd), m(mark), s(space) stop bits: 1, 1.5, 2 handshake: -(none), h(hardware:rts/cts), s(software:xon/xoff)

tttyb-mode	TTYB (baud rate, #bits, parity, #stop, handshake). Defaults to 9600,8,n,1,- . Fields, in left-to-right order, are: baud rate: 110, 300, 1200, 4800, 9600... data bits: 5, 6, 7, 8 stop bits: 1, 1.5, 2 parity: n(none), e(even), o(odd), m(mark), s(space) handshake: -(none), h(hardware:rts/cts), s(software:xon/xoff)
tttya-ignore-cd	If true , operating system ignores carrier-detect on TTYA. Defaults to true .
tttyb-ignore-cd	If true , operating system ignores carrier-detect on TTYB. Defaults to true .
tttya-rts-dtr-off	If true , operating system does not assert DTR and RTS on TTYA. Defaults to false .
tttyb-rts-dtr-off	If true , operating system does not assert DTR and RTS on TTYB. Defaults to false .
use-nvramrc?	If true , execute commands in NVRAMRC during system start-up. Defaults to false .
version2?	If true , hybrid (1.x/2.x) PROM comes up in version 2.x. Defaults to true .
watchdog-reboot?	If true , reboot after watchdog reset. Defaults to false .

EXAMPLES

The following example demonstrates the method for changing from one to two the number of megabytes of RAM that the system will test.

```
example# eeprom selftest-#megs
selftest-#megs=1

example# eeprom selftest-#megs=2

example# eeprom selftest-#megs
selftest-#megs=2
```

The following example demonstrates the method for setting the **auto-boot?** parameter to **true**.

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

When the **eeprom** command is executed in user mode, the parameters with a trailing question mark (?) need to be enclosed in double quotation marks (") to prevent the shell from interpreting the question mark. Preceding the question mark with an escape character (\) will also prevent the shell from interpreting the question mark.

example% eeprom "auto-boot?"=true

FILES

/dev/openprom device file
/usr/platform/*platform-name*/sbin/eeprom
platform-specific version of **eeprom**. To obtain *platform-name*, use **uname -i**.

SEE ALSO

passwd(1), sh(1), uname(1)

NAME	fdetach – detach a name from a STREAMS-based file descriptor
SYNOPSIS	fdetach <i>path</i>
DESCRIPTION	The fdetach command detaches a STREAMS-based file descriptor from a name in the file system. <i>path</i> is the path name of the object in the file system name space, which was previously attached (see fattach (3C)). The user must be the owner of the file or a user with the appropriate privileges. All subsequent operations on <i>path</i> will operate on the underlying file system entry and not on the STREAMS file. The permissions and status of the entry are restored to the state they were in before the STREAMS file was attached to the entry.
SEE ALSO	fattach (3C), fdetach (3C), streamio (7I) <i>STREAMS Programming Guide</i>

NAME	fdisk – create or modify fixed disk partition table
SYNOPSIS	<pre> fdisk [-o <i>offset</i>] [-s <i>size</i>] [-P <i>fill_patt</i>] [-S <i>geom_file</i>] [-w r d n I B t T g G R] [-F <i>fdisk_file</i>] [-W <i>creat_fdisk_file</i>] [-h] [-b <i>masterboot</i>] [-A <i>id:act:bhead:bsect:bcyl:ehead:esect:ecyl:rsect:numsect</i>] [-D <i>id:act:bhead:bsect:bcyl:ehead:esect:ecyl:rsect:numsect</i>] <i>rdevice</i> </pre>
AVAILABILITY	x86 SUNWcsr
DESCRIPTION	<p>This command is used to create and modify the partition table and install the master boot record that is put in the first sector of the fixed disk. This table is used by the first-stage bootstrap to identify parts of the disk reserved for different operating systems, and to identify the partition containing the second-stage bootstrap (the <i>active</i> Solaris partition). The <i>rdevice</i> argument must be used to specify the raw device associated with the fixed disk, for example, /dev/rdisk/c0t0d0p0.</p> <p>The program can operate in three different modes. The first is interactive mode. In interactive mode, the program displays the partition table as it exists on the disk, and then presents a menu allowing the user to modify the table. The menu, questions, warnings, and error messages are intended to be self-explanatory.</p> <p>In interactive mode, if there is no partition table on the disk, the user is given the options of creating a default partitioning or specifying the initial table values. The default partitioning allocates the entire disk for the Solaris x86 system and makes the Solaris system partition active. In either case, when the initial table is created, fdisk also writes out the first-stage bootstrap code along with the partition table.</p> <p>The second mode of operation is used for automated entry addition, entry deletion, or replacement of the entire fdisk table. This mode can add or delete an entry described on the command line. In this mode the entire fdisk table can be read in from a file replacing the original table. fdisk can also be used to create this file. There is a command line option that will cause fdisk to replace any fdisk table with the default of the whole disk for the Solaris x86 system.</p> <p>The third mode of operation is used for disk diagnostics. In this mode, a section of the disk can be filled with a user specified pattern, and mode sections of the disk can also be read or written.</p>
Menu Options	<p>The menu options for interactive mode given by the fdisk program are:</p> <p>Create a partition</p> <p>This option allows the user to create a new partition. The maximum number of partitions is 4. The program will ask for the type of the partition (SOLARIS, MS-DOS, UNIX, or other). It will then ask for the size of the partition as a percentage of the disk.</p>

The user may also enter the letter **c** at this point, in which case the program will ask for the starting cylinder number and size of the partition in cylinders. If a **c** is not entered, the program will determine the starting cylinder number where the partition will fit. In either case, if the partition would overlap an existing partition or will not fit, a message is displayed and the program returns to the original menu.

Change Active (Boot from) partition

This option allows the user to specify the partition where the first-stage bootstrap will look for the second-stage bootstrap, otherwise known as the *active* partition.

Delete a partition

This option allows the user to delete a previously created partition. Note that this will destroy all data in that partition.

Use the following options to include your modifications to the partition table at this time or to cancel the session without modifying the table:

Exit This option writes the new version of the table created during this session with *fdisk* out to the fixed disk, and exits the program.

Cancel This option exits without modifying the partition table.

OPTIONS

The following options apply to **fdisk**:

-S *geom_file* Set the label geometry to the content of the *geom_file*. The *geom_file* contains one specification line. Each line is delimited by a new-line character (**\n**). If the first character of a line is an asterisk (*), the line is treated as a comment. Each line is composed of entries that are position-dependent, are separated by "white space," and have the following format:

PCYL NCYL ACYL BCYL NHEADS NSECTORS SECTSIZ

where the entries have the following values:

PCYL This is the number of physical cylinders for the drive.

NCYL This is the number of usable cylinders for the drive.

ACYL This is the number of alt cylinders for the drive.

BCYL This is the number of offset cylinders for the drive (should be zero).

NHEADS The number of heads for this drive.

NSECTORS The number of sectors per track.

SECTSIZ The size in bytes of a sector.

-g Get the label geometry for disk and display on stdout (see the **-S** option for the format).

-G Get the physical geometry for disk and display on stdout (see the **-S** option for the format).

- n** Don't update **fdisk** table unless explicitly specified by another option. If no other options are used, **-n** will only write the master boot record to the disk. In addition, note that **fdisk** will not come up in interactive mode if the **-n** option is specified.
- I** Forgo device checks. This is used to generate a file image of what would go on a disk without using the device. Note that you must use **-S** with this option (see above).
- B** Default to one Solaris partition that uses the whole disk.
- t** Adjust incorrect slice table entries so that they will not cross partition table boundaries.
- T** Remove incorrect slice table entries that span partition table boundaries.
- b master_boot** Specify the file *master_boot* as the master boot program. The default master boot program is **/usr/platform/platform-name/lib/fs/ufs/mboot**. The platform name can be found using the **-i** option of **uname(1)**.
- R** Treat disk as read-only. This is for testing purposes.
- W fdisk_file** Create an **fdisk** file *fdisk_file* from disk table. This can be used with the **-F** option below.
- A id:act:bhead:bsect:bcyl:ehhead:esect:ecyl:rsect:numsect**
Add a partition as described by the argument (see the **-F** option below for the format).
- D id:act:bhead:bsect:bcyl:ehhead:esect:ecyl:rsect:numsect**
Delete a partition as described by the argument (see the **-F** option below for the format). Note that the argument must be an exact match or the entry will not be deleted!
- F fdisk_file** Use **fdisk** file *fdisk_file* to initialize table. The *fdisk_file* contains up to four specification lines. Each line is delimited by a new-line character (**\n**). If the first character of a line is an asterisk (*), the line is treated as a comment. Each line is composed of entries that are position-dependent, are separated by "white space" or colons, and have the following format:

id act bhead bsect bcyl ehhead esect ecyl rsect numsect
 where the entries have the following values:

<i>id</i>	This is the type of partition and the correct numeric values may be found in fdisk.h .
<i>act</i>	This is the active partition flag; 0 means not active and 128 means active.
<i>bhead</i>	This is the head where the partition starts. If this is set to 0 , fdisk will correctly fill this in from other information.
<i>bsect</i>	This is the sector where the partition starts. If this is set

- to **0**, **fdisk** will correctly fill this in from other information.
- b cyl* This is the cylinder where the partition starts. If this is set to **0**, **fdisk** will correctly fill this in from other information.
- e head* This is the head where the partition ends. If this is set to **0**, **fdisk** will correctly fill this in from other information.
- e sect* This is the sector where the partition ends. If this is set to **0**, **fdisk** will correctly fill this in from other information.
- e cyl* This is the cylinder where the partition ends. If this is set to **0**, **fdisk** will correctly fill this in from other information.
- r sect* The relative sector from the beginning of the disk where the partition starts. This must be specified and can be used by **fdisk** to fill in other fields.
- numsect* The size in sectors of this disk partition. This must be specified and can be used by **fdisk** to fill in other fields.
- r** Read from disk and write to stdout. See **-o** and **-s**, which specify the starting point and size of the operation.
- w** Write to disk and read from stdin. See **-o** and **-s**, which specify the starting point and size of the operation.
- P fill_patt** Fill disk with pattern *fill_patt*. *fill_patt* can be decimal or hex and is used as number for constant long word pattern. If *fill_patt* is #, then pattern is block # for each block. Pattern is put in each block as long words and fills each block (see **-o** and **-s**).
- o Offset** Block offset from start of disk. This option is used for **-P**, **-r**, and **-w**. Zero is assumed when this option is not used.
- s size** Number of blocks to perform operation on (see **-o**).
- d** Turn on verbose *debug* mode. This will cause **fdisk** to print its state on stderr as it is used. The output from this option should not be used with **-F**.
- h** Issue verbose message; message will list all options and supply an explanation for each.

DIAGNOSTICS

Most messages will be self-explanatory. The following may appear immediately after starting the program:

Fdisk: cannot open <device>

This indicates that the device name argument is not valid.

Fdisk: unable to get device parameters for device <device>

This indicates a problem with the configuration of the fixed disk, or an error in the fixed disk driver.

Fdisk: error reading partition table

This indicates that some error occurred when trying initially to read the fixed disk. This could be a problem with the fixed disk controller or driver, or with the configuration of the fixed disk.

Fdisk: error writing boot record

This indicates that some error occurred when trying to write the new partition table out to the fixed disk. This could be a problem with the fixed disk controller, the disk itself, the driver, or the configuration of the fixed disk.

FILES

/usr/platform/*platform-name*/lib/fs/ufs/mboot /dev/rdisk/c0t0d0p0

SEE ALSO

uname(1), fmthard(1M), prtvtoc(1M)

NAME	ff – list file names and statistics for a file system
SYNOPSIS	ff [-F <i>FSType</i>] [-V] [<i>generic_options</i>] [-o <i>specific_options</i>] <i>special</i> ...
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>ff prints the pathnames and inode numbers of files in the file system which resides on the special device <i>special</i>. Other information about the files may be printed using options described below. Selection criteria may be used to instruct ff to only print information for certain files. If no selection criteria are specified, information for all files considered will be printed (the default); the -i option may be used to limit files considered to those whose inodes are specified.</p> <p>Output is sorted in ascending inode number order. The default line produced by ff is:</p> <p style="padding-left: 40px;"><i>path-name i-number</i></p> <p>The maximum information the command will provide is:</p> <p style="padding-left: 40px;"><i>path-name i-number size uid</i></p>
OPTIONS	<p>-F Specify the <i>FSType</i> on which to operate. The <i>FSType</i> should either be specified here or be determinable from /etc/vfstab by matching the <i>special</i> with an entry in the table, or by consulting /etc/default/fs.</p> <p>-V Echo the complete command line, but do not execute the command. The command line is generated by using the options and arguments provided by the user and adding to them information derived from /etc/vfstab. This option may be used to verify and validate the command line.</p> <p><i>generic_options</i> Options that are supported by most <i>FSType</i>-specific modules of the command. The following options are available:</p> <ul style="list-style-type: none"> -I Do not print the i-node number after each path name. -l Generate a supplementary list of all path names for multiply-linked files. -p prefix The specified <i>prefix</i> will be added to each generated path name. The default is '.' (dot). -s Print the file size, in bytes, after each path name. -u Print the owner's login name after each path name. -a n Select if the file has been accessed in <i>n</i> days. -m n Select if the file has been written or created in <i>n</i> days. -c n Select if file's status has been changed in <i>n</i> days. -n file Select if the file has been modified more recently than the argument <i>file</i>.

- i** *i-node-list* Generate names for only those i-nodes specified in *i-node-list*. *i-node-list* is a list of numbers separated by commas (with no intervening spaces).
- o** Specify *FSType*-specific options in a comma separated (without spaces) list of suboptions and keyword-attribute pairs for interpretation by the *FSType*-specific module of the command.
- FILES**
- /etc/default/fs** default local file system type. Default values can be set for the following flags in **/etc/default/fs**. For example: LOCAL=ufs
- LOCAL:** The default partition for a command if no *FSType* is specified.
- /etc/vfstab** list of default parameters for each file system
- SEE ALSO** **find(1)**, **ncheck(1M)**, **stat(2)**, **vfstab(4)**
Manual pages for the *FSType*-specific modules of **ff**.
- NOTES**
- This command may not be supported for all *FSTypes*.
The **-a**, **-m**, and **-c** flags examine the **st_atime**, **st_mtime**, and **st_ctime** fields of the **stat** structure respectively. (See **stat(2)**.)

NAME	ff_ufs – list file names and statistics for a ufs file system
SYNOPSIS	ff -F ufs [<i>generic_options</i>] [-o a,m,s] <i>special</i> . . .
DESCRIPTION	ff prints the pathnames and inode numbers of files in the file system which resides on the special device <i>special</i> . ff is described in ff(1M) ; ufs-specific options are described below.
OPTIONS	-o Specify ufs file system specific options. The options available are: a Print the '.' and '..' directory entries. m Print mode information. s Print only special files and files with set-user-ID mode.
SEE ALSO	find(1) , ff(1M) , ncheck(1M)

NAME	firmware – bootable firmware programs and firmware commands
AVAILABILITY	SPARC
DESCRIPTION	<p>Between the time most computers are turned on and the boot program is loaded to bootstrap the machine, the computer is in an operating state known as the firmware state. In the firmware state, a small program in non-volatile memory is running on the machine, and the user can perform certain system operations usually unavailable from single- or multi-user operating states.</p> <p>There are two basic kinds of firmware operations:</p> <p style="padding-left: 40px;">Running firmware commands. These commands might include commands for displaying the Equipped Device Table, performing a system memory dump, displaying the firmware version, creating a floppy key, etc. These commands are executed by the firmware program.</p> <p style="padding-left: 40px;">Running bootable programs. These include the operating system and other bootable programs (such as a program to fill the Equipped Device Table). These programs are located in the /stand file system. When a bootable program is requested from firmware, the firmware program loads and executes the program, passing control of the system to the bootable program.</p> <p>Some firmware programs, allow you to request the configuration of a new bootable operating system from firmware by specifying the name of a configuration file (usually /stand/system) as the name of the program to boot; see system(4).</p> <p>See the hardware guide that accompanies your computer for descriptions of the firmware commands and programs available with your machine.</p>
SEE ALSO	<p>system(4)</p> <p><i>Writing FCode Programs</i></p> <p><i>OpenBoot Command Reference Manual</i></p>
WARNINGS	The firmware program typically does not know if a requested program is bootable or not; requesting a program that is not bootable from firmware can lead to unpredictable results.

NAME	fmthard – populate VTOC on hard disks
SPARC SYNOPSIS	fmthard -d <i>data</i> -n <i>volume_name</i> -s <i>datafile</i> [-i] /dev/rdisk/c?[t?]d?s2
x86 SYNOPSIS	fmthard -d <i>data</i> -n <i>volume_name</i> -s <i>datafile</i> [-i] [-p <i>pboot</i>] [-b <i>bootblk</i>] /dev/rdisk/c?[t?]d?s2
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>The fmthard command updates the VTOC (Volume Table of Contents) on hard disks and, on x86 systems, adds boot information to the Solaris fdisk partition. One or more of the options -s <i>datafile</i>, -d <i>data</i>, or -n <i>volume_name</i>, must be used to request modifications to the disk label. To print disk label contents, see prtvtoc(1M). The /dev/rdisk/c?[t?]d?s2 file must be the character special file of the device where the new VTOC is to be installed. On x86 systems, fdisk(1M) must be run on the drive before fmthard.</p> <p>If you are using an x86 system, note that the term “partition” in this page refers to <i>slices</i> within the x86 fdisk partition on x86 machines. Do not confuse the partitions created by fmthard with the partitions created by fdisk.</p>
OPTIONS	<p>The following options apply to fmthard:</p> <p>-i This option allows the command to create the desired VTOC table, but prints the information to standard output instead of modifying the VTOC on the disk.</p> <p>-d <i>data</i> The <i>data</i> argument of this option is a string representing the information for a particular partition in the current VTOC. The string must be of the format <i>part:tag:flag:start:size</i> where <i>part</i> is the partition number, <i>tag</i> is the ID TAG of the partition, <i>flag</i> is the set of permission flags, <i>start</i> is the starting sector number of the partition, and <i>size</i> is the number of sectors in the partition. See the description of the <i>datafile</i> below for more information on these fields.</p> <p>-n <i>volume_name</i> This option is used to give the disk a <i>volume_name</i> up to 8 characters long.</p> <p>-s <i>datafile</i> This option is used to populate the VTOC according to a <i>datafile</i> created by the user. If the <i>datafile</i> is "-", fmthard reads from standard input. The <i>datafile</i> format is described below. This option causes all of the disk partition timestamp fields to be set to zero.</p> <p>Every VTOC generated by fmthard will also have partition 2, by convention, that corresponds to the whole disk. If the input in <i>datafile</i> does not specify an entry for partition 2, a default partition 2 entry will be created automatically in VTOC with the tag V_BACKUP and size equal to the full size of the disk.</p> <p>The <i>datafile</i> contains one specification line for each partition, starting with partition 0. Each line is delimited by a new-line character (\n). If the first character of a line is an asterisk (*), the line is treated as a comment.</p>

Each line is composed of entries that are position-dependent, separated by "white space" and having the following format:

```
partition tag flag starting_sector size_in_sectors
```

where the entries have the following values.

<i>partition</i>	The partition number: 0-15 decimal or 0x0-0xf hexadecimal.
<i>tag</i>	The partition tag: a two-digit hex number. The following are reserved codes: 0x00 (V_UNASSIGNED), 0x01 (V_BOOT), 0x02 (V_ROOT), 0x03 (V_SWAP), 0x04 (V_USR), 0x05 (V_BACKUP), 0x06 (V_STAND), 0x07 (V_VAR) and 0x08 (V_HOME).
<i>flag</i>	The flag allows a partition to be flagged as unmountable or read only, the masks being: V_UNMNT 0x01 , and V_RDONLY 0x10 . For mountable partitions use 0x00 .
<i>starting sector</i>	The sector number (decimal) on which the partition starts.
<i>size in sectors</i>	The number (decimal) of sectors occupied by the partition.

Note that you can save the output of a **prtvtoc** command to a file, edit the file, and use it as the *datafile* argument to the **-s** option.

x86 Options

The functionality provided by the following two x86 options is also provided by **installboot**(1M). Because the functionality described here may be removed in future versions of **fmthard**, you should use **installboot** to install boot records. The following options currently apply to **fmthard**:

-p <i>pboot</i>	This option allows the user to override the default partition boot file, /usr/platform/platform-name/lib/fs/ufs/pboot . The partition boot file is platform dependent, where <i>platform-name</i> can be determined using the -i option to uname (1).
-b <i>bootblk</i>	This option allows the user to override the default <i>bootblk</i> file, /usr/platform/platform-name/lib/fs/ufs/bootblk . The boot block file is platform dependent, where <i>platform-name</i> can be determined using the -i option to uname (1).

SEE ALSO

x86 Only

uname(1), **format**(1M), **prtvtoc**(1M)
fdisk(1M), **installboot**(1M)

NOTES

Special care should be exercised when overwriting an existing VTOC, as incorrect entries could result in current data being inaccessible. As a precaution, save the old VTOC. **fmthard** cannot write a disk label on an unlabeled disk. Use **format**(1M) for this purpose.

NAME	fncheck – check for consistency between FNS data and NIS+ data
SYNOPSIS	fncheck [-r] [-s] [-u] [-t type] [<i>domain_name</i>]
DESCRIPTION	<p>fncheck is used for checking for inconsistencies between FNS username or hostname contexts and the contents of the corresponding NIS+ <i>passwd.org_dir</i> or <i>hosts.org_dir</i> tables, respectively, in the NIS+ domain <i>domain_name</i>. If <i>domain_name</i> is omitted, the domain name of the current machine is used. By default (in the absence of the -r and -s options), the following inconsistencies are displayed:</p> <ul style="list-style-type: none"> • items that appear only in the FNS context but do not appear in the NIS+ table, • items that appear only in the NIS+ table but do not appear in the FNS context.
OPTIONS	<p>-r Display only items that appear in the FNS context but do not appear in the corresponding NIS+ table.</p> <p>-s Display items that appear in the NIS+ table but do not appear in the corresponding FNS context.</p> <p>-u Update the FNS context based on information in the corresponding NIS+ table. If the -r option is used, items that appear only in the FNS context are removed from the FNS context. If the -s option is used, items that appear only in the NIS+ table are added to the FNS context. If neither -r or -s are specified, items are added and removed from the FNS context to make it consistent with the corresponding NIS+ table.</p> <p>-t type Specify the type of context to check. <i>type</i> can be either hostname or username. If this option is omitted, both hostname and username contexts are checked. If <i>type</i> is hostname, the FNS hostname context is checked against the NIS+ <i>hosts.org_dir</i> table. If <i>type</i> is username, the FNS username context is checked against the NIS+ <i>passwd.org_dir</i> table.</p>
USAGE	<p>Although fncheck can be used to add users and hosts to the username and hostname contexts as new users and hosts are added to NIS+, that is not its intended purpose. fncheck is an expensive operation because it makes complete comparisons of the NIS+ table and the corresponding FNS context. When a user or host is added or removed from NIS+ using admintool (see admintool(1M)), it automatically updates the appropriate FNS contexts.</p>
SEE ALSO	admintool(1M) , fncreate(1M) , fndestroy(1M) , nis(1) , fns(5) , fns_policies(5)

NAME	fncreate – Create an FNS context
SYNOPSIS	fncreate -t <i>context_type</i> [-f <i>input_file</i>] [-o] [-r <i>reference_type</i>] [-s] [-v] [-D] <i>composite_name</i>
DESCRIPTION	<p>fncreate creates an FNS context of type <i>context_type</i>, where a <i>context_type</i> must be one of org, hostname, host, username, user, service, fs, site, nsid, or generic. It takes as the last argument a composite name for the context to be created. In addition to creating the context named, fncreate also creates subcontexts of the named context using FNS Policies of what types of contexts should be bound in those contexts. See fns_policies(5).</p> <p>fncreate creates NIS+ tables and directories in the NIS+ hierarchy. See fns(5) for more information on the necessary NIS+ credentials and the use of the environment variable NIS_GROUP when using fncreate and other FNS commands.</p>
OPTIONS	<p>-t org Create organization context, and default subcontexts, for an existing NIS+ domain. <i>composite_name</i> is of the form <i>org/domain/</i> where domain is an NIS+ domain name. An empty domain name indicates the creation of the organization context for the root NIS+ domain; otherwise, the domain name names the corresponding NIS+ domain. domain can be either the fully-qualified NIS+ domain name — dot (‘.’) terminated — or the NIS+ domain name named relative to the NIS+ root domain.</p> <p>For example, the following creates the root organization context and its subcontexts for the NIS+ root domain Wiz.COM:</p> <pre>eg% fncreate -t org org//</pre> <p>The same thing could have been achieved using the following command:</p> <pre>eg% fncreate -t org org/Wiz.COM./</pre> <p>Typically, this is the first FNS context created.</p> <p>To create the organization context for a subdomain of Wiz.COM., execute either of the following commands:</p> <pre>eg% fncreate -t org org/sales/</pre> <p>or</p> <pre>eg% fncreate -t org org/sales.Wiz.COM./</pre> <p>Note that if the corresponding NIS+ domain does not exist, fncreate fails. See nissetup(1M) for setting up a NIS+ domain.</p> <p>A ctx_dir directory is created under the directory of the organization named. Its immediate subcontexts host, user and service and their subcontexts, including a context for every host entry in the corresponding <i>hosts.org_dir</i> NIS+ table and every user entry in the <i>passwd.org_dir</i> NIS+ table, are created unless the option -o is specified. Bindings for these subcontexts are recorded under the organization context.</p>

- t hostname** Create a hostname context in which atomic **host** names can be bound, and bind the reference of the context to *composite_name*. If the suffix of *composite_name* is **host/**, the hostname context created is also bound to the composite name with this suffix replaced by **_host/**, and vice versa (i.e. if a composite name with a **_host/** suffix was supplied, a binding would be created for **host/**). Also create a host context for every host entry in the corresponding *hosts.org_dir* NIS+ table unless either options **-o** or **-f** is specified.
eg% fncreate -t hostname org/sales/host/
Typically, a hostname context need not be created explicitly since it is created by default, as a subcontext under **org**.
- t host** Create a host context for a specific host, and its **service** and **fs** subcontexts, and bind the reference of the context to *composite_name*. For example, the following creates a host context and **service** and **fs** subcontexts for host *sylvan*.
eg% fncreate -t host org/sales/host/sylvan/
- t username** Create a username context in which atomic **user** names can be bound, and bind the reference of the context to *composite_name*. If the suffix of *composite_name* is **user/**, the username context created is also bound to the composite name with this suffix replaced by **_user/**, and vice versa. Also create a user context for every user entry in the corresponding *passwd.org_dir* NIS+ table unless either the options **-o** or **-f** is specified.
eg% fncreate -t username org/sales/user/
Typically, a username context need not be created explicitly since it is created by default, as a subcontext under **org**.
- t user** Create a user context for a specific user, and its **service** and **fs** subcontexts, and bind the reference of the context to *composite_name*. For example, the following creates a user context and **service** and **fs** subcontexts for user *jsmith*.
eg% fncreate -t user org/sales/user/jsmith/
- t service** Create a service context in which slash-separated left-to-right service names can be bound, and bind the reference of the context to *composite_name*. If the suffix of *composite_name* is **service/**, the service context created is also bound to the composite name with this suffix replaced by **_service/**, and vice versa. Typically, a service context need not be created explicitly since it is created by default, as a subcontext under **org**, **host**, or **user** contexts.
- t fs** Create a file system context for a user or host, and bind the reference of the context to *composite_name*. The composite name must be the name of a host or a user, with either **fs/** or **_fs/** appended to it. If the suffix of *composite_name* is **fs/**, the file system context created is also bound to the composite name with this suffix replaced by **_fs/**, and vice versa.
Typically, a file system context need not be created explicitly since it is

created by default, as a subcontext of a user or host context.

The file system context of a user is the user's home directory as stored in the NIS+ table **passwd.org_dir**. The file system context of a host is the set of NFS file systems that the host exports.

Use the **fncreate_fs(1M)** command to create file system contexts for organizations and sites, or to create file system contexts other than the defaults for users and hosts.

- t **site** Create a site context in which dot-separated right-to-left site names can be bound, and a service subcontext, and bind the reference of the context to *composite_name*. If the suffix of *composite_name* is **site/**, the hostname context created is also bound to the composite name with this suffix replaced by **_site/**, and vice versa. Typically, a site context is created at the same level as the **org** context and is used for creating a geographical namespace that complements the organizational namespace of an enterprise.
- t **nsid** Create a context in which namespace identifiers can be bound. This context has a flat namespace, in which only atomic names can be bound. An example of such a context is the context to which the name **site/east/** is bound. This context can have the following bindings: **site/east/host**, **site/east/user**, and **site/east/service**.
- t **generic** Create a generic context in which slash-separated left-to-right names can be bound, and bind the reference of the context to *composite_name*. The option **-r** can be used to specify the reference type to be associated with the context. If the **-r** option is omitted, the reference type used is the reference type of the parent context if the parent context is a generic context; otherwise, the reference type is **onc_fn_generic**.
- f Create a context for every user or host listed in **input_file**. This option is only applicable when used with the **-t username** or **-t hostname** options. The format of the file is an atomic user name or host name per line. This option is used to create contexts for a subset of the users/hosts found in the corresponding NIS+ *passwd.org_dir* or *hosts.org_dir* tables, respectively. If this option is omitted, **fncreate** creates a context for every user/host found in the corresponding NIS+ *passwd.org_dir* or *hosts.org_dir* tables, respectively.
- o Only the context named by *composite_name* is created; no subcontexts are created. When this option is omitted, subcontexts are created according to the FNS Policies for the type of the new object.
- r Use *reference_type* as the reference type of the generic context being created. This option can only be used with the **-t generic** option.
- s Create the context and bind it in to supercede any existing binding associated with *composite_name*. If this option is omitted, **fncreate** fails if *composite_name* is already bound.

- v Information about the creation of a context is displayed as each context is created.
- D Information about the creation of a context, and corresponding NIS+ directories and tables, is displayed as each context is created.

RETURN VALUES

- 1 This command returns **0** if it is successful. It returns **1** if it is not successful.

EXAMPLES

The following examples illustrate creation of a host context in the root organization and a user context in a sub-organization.

Create a context, and subcontexts, for the root organization:

```
eg% fncreate -t org org//
```

It causes the following commands to be invoked automatically:

```
eg% fncreate -t service org//service/
```

```
eg% fncreate -t hostname org//host/
```

```
eg% fncreate -t username org//user/
```

Create a context, and subcontexts, for host *sylvan*:

```
eg% fncreate -t host org//host/sylvan/
```

It causes the following commands to be invoked automatically:

```
eg% fncreate -t service org//host/sylvan/service/
```

```
eg% fncreate -t fs org//host/sylvan/fs/
```

Create a context, and subcontexts, associated with a sub-organization *dct*:

```
eg% fncreate -t org org/dct/
```

It causes the following commands to be invoked automatically:

```
eg% fncreate -t service org/dct/service/
```

```
eg% fncreate -t hostname org/dct/host/
```

```
eg% fncreate -t username org/dct/user/
```

Create a context, and subcontexts, for user *msmith*:

```
eg% fncreate -t user org/dct/user/msmith/
```

It causes the following command to be invoked automatically:

```
eg% fncreate -t service org/dct/user/msmith/service/
```

```
eg% fncreate -t fs org/dct/user/msmith/fs/
```

The following examples create service contexts:

```
eg% fncreate -t service org/dct/service/fax
```

```
eg% fncreate -t service org/dct/service/fax/classA
```

SEE ALSO

fncheck(1M), **fncreate_fs(1M)**, **fndestroy(1M)**, **nis(1)**, **nissetup(1M)**, **xfn(3N)**, **fns(5)**, **fns_policies(5)**, **fns_references(5)**

NAME	fncreate_fs – create FNS file system contexts
SYNOPSIS	fncreate_fs [-r] [-v] -f <i>input_file</i> <i>composite_name</i> fncreate_fs [-r] [-v] <i>composite_name</i> [<i>mount_options</i>] [<i>mount_location ...</i>]
DESCRIPTION	The fncreate_fs command creates or updates the FNS file system context named by <i>composite_name</i> . A description of the context's bindings is provided in <i>input_file</i> if the first form of the command is used, or is given on the command line if the second form is used.
OPTIONS	<p>-r Replace the bindings in the context named by <i>composite_name</i> with only those specified in the input. This is equivalent to destroying the context (and, recursively, its subcontexts), and then running fncreate_fs without this option. This option should be used with care.</p> <p>-v Verbose. Display information about the contexts being created and modified.</p> <p>-f <i>input_file</i> Read input from <i>input_file</i>. If <i>input_file</i> is '-' (hyphen), read from standard input instead.</p>
USAGE	<p>The fncreate_fs command populates the file system portions of the FNS namespace. The automounter (see automount(1M)) will then "mount" the FNS namespace under /xfn. The directory with the FNS name org/engineering/fs, for example, can be found on the file system as /xfn/org/engineering/fs.</p> <p>The format of the input to fncreate_fs is similar, but not identical, to the format of indirect automount maps. Differences are enumerated in the NOTES section below.</p>
Input File Format	<p>The input file supplies the names and values to be bound in the context of <i>composite_name</i>. Its format is a sequence of lines of the form:</p> <p style="padding-left: 40px;"><i>name</i> [-<i>options</i>] [<i>location ...</i>]</p> <p>For each such entry, a reference to the <i>location(s)</i> and the corresponding <i>options</i> is bound to the name <i>composite_name/name</i>. The <i>name</i> field may be a simple atomic name, a slash-separated hierarchical name, or '.' (period). If it is '.' then the reference is bound directly to <i>composite_name</i>. The <i>name</i> field must not begin with a slash.</p> <p>The <i>location</i> field specifies the host or hosts that serve the files for <i>composite_name/name</i>. In the case of a simple NFS mount, <i>location</i> takes the form:</p> <p style="padding-left: 40px;"><i>host</i>: <i>path</i></p> <p>where <i>host</i> is the name of the host from which to mount the file system, and <i>path</i> is the path name of the directory to mount.</p> <p>The <i>options</i> field is a comma-separated list of the mount options to use when mounting the location bound to <i>composite_name/name</i>. These options also apply to any subcontexts of <i>composite_name/name</i> that do not specify their own mount options. If <i>options</i> is given but <i>location</i> is not, the options apply to subcontexts only.</p>

	<p>If neither <i>options</i> nor a <i>location</i> is given, then no reference is bound to <i>composite_name/name</i>. Any existing reference is unbound.</p> <p>A single logical line may be continued across multiple input lines by escaping the new-line with a '\ ' (backslash). Comments begin with a '#' that is either at the beginning of a line or is prefixed by whitespace, and end at the end of the line.</p>
Command-line Input	<p>If no <i>input_file</i> is specified on the command line, then the <i>options</i> and <i>location</i> fields given on the command line are bound directly to <i>composite_name</i>. This is equivalent to providing a one-line input file with a '.' in the <i>name</i> field.</p>
Multiple Locations	<p>Multiple <i>location</i> fields may be specified for file systems that are exported from multiple, functionally-equivalent locations. If several locations in the list share the same path name, they may be combined using a comma-separated list of host names:</p> <p style="padding-left: 40px;"><i>host1, host2, ... : path</i></p> <p>The hosts may be weighted, with the weighting factor appended to the host name as a non-negative integer in parentheses: the lower the number, the more desirable the server. The default weighting factor is 0 (most desirable). In the example:</p> <p style="padding-left: 40px;">alpha,bravo,charlie(1),delta(2):/usr/man</p> <p>hosts alpha and bravo are the most desirable; host delta, the least.</p> <p>See the USAGE section of automount(1M) for additional information on how the automounter interprets the location field.</p>
Variable Substitution	<p>Variable names, prefixed by '\$', may be used with the <i>options</i> or <i>location</i> fields. For example, a <i>location</i> may be given as:</p> <p style="padding-left: 40px;">svr1:/export/\$CPU</p> <p>The automounter will substitute client-specific values for these variables when mounting the corresponding file systems. In the above example, \$CPU is replaced by the output of uname-p; for example, "sparc". See the USAGE section of automount(1M) for more information on how the automounter treats variable substitution.</p>
Alternate Input Format	<p>For additional compatibility with automount maps (see automount(1M)), the following input format is accepted:</p> <pre style="padding-left: 40px;"> name [options] [location ...] \ /offset1 [options1] location1 ... \ /offset2 [options2] location2 ... \ ... </pre> <p>where each <i>offset</i> field is a slash-separated hierarchy. This is interpreted as being equivalent to:</p> <pre style="padding-left: 40px;"> name [options] [location ...] name/offset1 [options1] location1 ... name/offset2 [options2] location2 </pre>

(the first line being omitted if both *options* and *location* are omitted).

This format is for compatibility only; it provides no additional functionality. Its use is deprecated.

EXAMPLES

The following examples illustrate the use of the **fcreate_fs** command.

The call:

```
example% cat input1
src      -ro      svr1:/export/src
dist     -ro      svr2,svr3:/export/dist
example% fcreate_fs -f input1 org/engineering/fs
```

creates a file system context for the engineering organization. It specifies that **org/engineering/fs/src** is a read-only NFS mount from server **svr1**, and that **org/engineering/fs/dist** is a read-only NFS mount from either **svr2** or **svr3**.

Once this is done, there are several equivalent ways to create the engineering organization's **src/cmd** context. It could be done using the composite name **org/engineering/fs**:

```
example% cat input2
src/cmd      svr1:/export/cmd
example% fcreate_fs -f input2 org/engineering/fs
```

Equivalently, it could be done using the composite name **org/engineering/fs/src**:

```
example% cat input3
cmd          svr1:/export/cmd
example% fcreate_fs -f input3 org/engineering/fs/src
```

The same results could also be achieved by:

```
example% fcreate_fs org/engineering/fs/src/cmd svr1:/export/cmd
```

Note that **cmd** will also be mounted read-only, since it is a subcontext of **src** and does not have mount options of its own.

In the first example of this section, the **-ro** mount option was specified for each entry in the input file. It could instead have been specified only once:

```
example% cat input4
.      -ro
src      svr1:/export/src
dist     svr2,svr3:/export/dist
example% fcreate_fs -f input4 org/engineering/fs
```

The **-ro** option here applies to all bindings in the context **org/engineering/fs** and any of its subcontexts. In particular, it also applies to the **cmd** context from the above examples.

The following will change the NFS server for the **src** context:

```
example% fcreate_fs org/engineering/fs/src svr4:/export/src
```

Had the **-r** option been used, the **cmd** subcontext would have been destroyed as well:

```
example% fncreate_fs -r org/engineering/fs/src svr4:/export/src
```

Only the FNS context is destroyed. The **/export/cmd** directory on **svr1** is not affected.

The file system contexts of users and hosts are not usually created by **fncreate_fs** (see the **NOTES** section below). The defaults set by **fncreate**, however, may be overridden. For example, the call:

```
example% fncreate_fs user/jane/fs svr1:/export/home/jane
```

sets Jane's file system to be an NFS mount from **svr1**.

SEE ALSO

fnbind(1), **fnlist(1)**, **fnlookup(1)**, **fnunbind(1)**, **automount(1M)**, **fncreate(1M)**, **fndestroy(1M)**, **nis+(1M)**, **fns(5)**, **fns_policies(5)**

NOTES

The **fncreate_fs** command affects the FNS file system namespace only. It does not have any effect on the servers that export the files and directories from which the namespace is constructed. Destroying an FNS context does not remove any files on any server.

FNS policies specify that file system contexts are bound after the namespace identifier **fs** in composite names (see **fns_policies(5)**). Therefore, *composite_name* must contain an **fs**. The alias **_fs** may be used in place of **fs**.

The context named by the components of *composite_name* preceding **fs** must exist prior to the call to **fncreate_fs**, since **fncreate_fs** creates *only* file system contexts.

Default file system contexts for hosts and users are generally created by the command **fncreate(1M)**. These defaults may be overridden using **fncreate_fs**. Overriding a host's default file system context is unlikely to make sense.

The input file format is similar to the format of indirect automount maps (see **automount(1M)**). The differences are:

- the *name* field may be hierarchical, and may be ':';
- there are no included maps or special maps;
- there may be entries with neither options nor locations;
- the characters '*' and '&' have no special meaning.

The **fncreate_fs** command modifies certain NIS+ tables. The executing process must have the necessary NIS+ credentials to do so. See **fns(5)** and **nis+(1M)** for more information.

NAME	fncreate_printer – create new printers in the FNS namespace
SYNOPSIS	fncreate_printer [-sv] <i>compositename printername printeraddr</i> [<i>printeraddr ...</i>] fncreate_printer [-sv] [-f filename] <i>compositename</i>
DESCRIPTION	<p>fncreate_printer creates a new printer for an <i>organization</i>, <i>user</i>, <i>host</i>, or <i>site</i> object. <i>compositename</i> is the FNS name of the object. fncreate_printer uses <i>printername</i> to name the new printer and binds it to an FNS reference constructed from the set of <i>printeraddrs</i>. fncreate_printer may also be used to add new <i>printeraddrs</i> for an existing <i>printername</i>. The command supports creating a set of printers as listed in the file <i>filename</i>.</p> <p>The new printer is created with the FNS name <code><compositename>/service/printer/<printername></code>. If the intermediate service or printer names do not exist, their FNS contexts are also created by this command. Normally, these intermediate contexts would be created by an administrative script that uses fncreate(1M), and is run at the time a new FNS organization is set up. The reference bound to the FNS printer name is of type onc_printers and is constructed from the set of <i>printeraddrs</i>. A <i>printeraddr</i> is of the form <code><addresstype>=<address></code>. See the examples below for currently supported address types and address strings.</p> <p>An FNS printer name is accepted as a valid printer name by lp(1), lpstat(1), cancel(1), lpmove(1M), lpr(1B), lpq(1B), and lprm(1B).</p> <p>The <i>printername</i> argument may be a slash-separated name. In this case, prior to creating the printer denoted by the “leaf” name, this command will create printer(s) for the intermediate node(s) if they do not already exist. See examples below.</p> <p>fncreate_printer creates NIS+ entries in the NIS+ hierarchy for FNS. See fns(5) for more information on the necessary NIS+ credentials and the use of the environment variable NIS_GROUP when using fncreate and other FNS commands.</p>
OPTIONS	<p>-s The new <i>address</i> supersedes an existing address with the same <i>addresstype</i>, if any, for <code><compositename>/service/printer/<printername></code>. If this option is omitted, it appends the <i>printeraddr</i> to an existing reference, or creates a new reference using <i>printeraddr</i> for the printer.</p> <p>-v Displays information about individual printers as they are created.</p> <p>-f filename Use <i>filename</i> to obtain a list of printers to be created. If this option is omitted, <code>/etc/printers.conf</code> is used as the input file.</p>
OPERANDS	<p><i>filename</i> The file that contains a list of printers to be created. This file uses the same format as <code>/etc/printers.conf</code>.</p> <p><i>printername</i> The name of the new printer created.</p> <p><i>printeraddr</i> An address to be associated with the printer name.</p> <p><i>compositename</i> The FNS name for the org, host, user, or site object for which the new printers are created.</p>

EXAMPLES

The following examples illustrate creating a set of printers under an organization, a printer for a user, and a printer associated with a hierarchical printer name for a site, respectively.

Example 1

Create printers for an organization:

```
example% fncreate_printer org/marketing
```

This causes the creation of a printer for every entry listed in the `/etc/printers.conf` file on the system where the command is executed. The printers thus created are bound under the organization's printer context, `org/marketing/service/printer`.

Example 2

Create a printer named `ps` for user `jsmith` and associate it with the `killtree` printer served by the print server `paperwaster`:

```
example% fncreate_printer -s usr/jsmith ps \
      bsdaddr=paperwaster,killtree
```

This causes `jsmith's ps` printer to be associated with the `killtree` printer on the server `paperwaster`, overwriting the existing address, if any, of type `bsdaddr`.

Example 3

Create a printer with the hierarchical name `color/fast` under a site:

```
example% fncreate_printer site/bldg14/northwing color/fast \
      bsdaddr=paperwaster,laser
```

This causes the printer named `site/bldg14/northwing/service/printer/color/fast` to be associated with the `laser` printer on server `paperwaster`. If the intermediate printer name `site/bldg14/northwing/service/printer/color` does not exist, it will also be created and associated with the same printer. If the printer name `site/bldg14/northwing/service/printer/color/fast` already exists and has an address of type `bsdaddr` associated with it, this command will fail.

EXIT STATUS

This command returns `0` on success and `1` if it is not successful.

SEE ALSO

`cancel(1)`, `lp(1)`, `lpr(1B)`, `lprm(1B)`, `lpq(1B)`, `lpstat(1)`, `fncreate(1M)`, `lpmove(1M)`, `fns(5)`

NAME	fndestroy – Destroy an FNS context
SYNOPSIS	fndestroy <i>composite_name</i>
DESCRIPTION	fndestroy removes the context bound to <i>composite_name</i> . The context is not removed if there are subcontexts associated with <i>composite_name</i> .
EXAMPLES	The command eg% fndestroy user/jsmith/ destroys the context named by user/jsmith/ and removes the binding of jsmith from the context user/ . This command fails if the context user/jsmith/ contains subcontexts, or if the invoker does not have the NIS+ credentials required to delete the NIS+ tables that store the user's bindings. See fns(5) .
SEE ALSO	fncreate(1M) , fnlist(1) , fnlookup(1) , fnunbind(1) , fns(5) , fns_policies(5)

NAME	format – disk partitioning and maintenance utility
SYNOPSIS	format [-f <i>command-file</i>] [-l <i>log-file</i>] [-x <i>data-file</i>] [-d <i>disk-name</i>] [-t <i>disk-type</i>] [-p <i>partition-name</i>] [-s] [-m] [-M] [-e] [<i>disk-list</i>]
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>format enables you to format, label, repair and analyze disks on your system. Unlike previous disk maintenance programs, format runs under SunOS. Because there are limitations to what can be done to the system disk while the system is running, format is also supported within the memory-resident system environment. For most applications, however, running format under SunOS is the more convenient approach.</p> <p>format first uses the disk list defined in <i>data-file</i> if the -x option is used. format then checks for the FORMAT_PATH environment variable, a colon-separated list of filenames and/or directories. In the case of a directory, format searches for a file named format.dat in that directory; a filename should be an absolute pathname, and is used without change. format adds all disk and partition definitions in each specified file to the working set. Multiple identical definitions are silently ignored. If FORMAT_PATH is not set, the path defaults to /etc/format.dat.</p> <p><i>disk-list</i> is a list of disks in the form c?t?d? or /dev/rdisk/c?t?d?s?. With the latter form shell wildcard specifications are supported. For example, specifying /dev/rdisk/c2* will cause format to work on all drives connected to controller c2 only. If no <i>disk-list</i> is specified, format lists all the disks present in the system.</p>
OPTIONS	<p>The following options are supported:</p> <p>-f <i>command-file</i> Take command input from <i>command-file</i> rather than the standard input. The file must contain commands that appear just as they would if they had been entered from the keyboard. With this option, format does not issue continue? prompts; there is no need to specify y(es) or n(o) answers in the <i>command-file</i>. In non-interactive mode, format does not initially expect the input of a disk selection number. The user must specify the current working disk with the -d <i>disk-name</i> option when format is invoked, or specify disk and the disk selection number in the <i>command-file</i>.</p> <p>-l <i>log-file</i> Log a transcript of the format session to the indicated <i>log-file</i>, including the standard input, the standard output and the standard error.</p> <p>-x <i>data-file</i> Use the list of disks contained in <i>data-file</i>.</p> <p>-d <i>disk-name</i> Specify which disk should be made current upon entry into the program. The disk is specified by its logical name (for instance, -c0t1d0). This can also be accomplished by specifying a single disk in the disk list.</p> <p>-t <i>disk-type</i> Specify the type of disk which is current upon entry into the program. A disk's type is specified by name in the data file. This option can only be used if a disk is being made current as described above.</p>

- p** *partition-name*
Specify the partition table for the disk which is current upon entry into the program. The table is specified by its name as defined in the data file. This option can only be used if a disk is being made current, and its type is either specified or available from the disk label.
- e**
Enable SCSI expert menu. Note this option is not recommended for casual use.
- s**
Silent. Suppress all of the standard output. Error messages are still displayed. This is generally used in conjunction with the **-f** option.
- m**
Enable extended messages. Provides more detailed information in the event of an error.
- M**
Enable extended and diagnostic messages. Provides extensive information on the state of an SCSI device's mode pages, during formatting.

USAGE

The **format** utility's main menu items allow you to do the following tasks:

- analyze** Run read, write, and compare tests.
- backup** Search for backup labels.
- current** Display the device name, manufacturer and model of the current disk.
- defect** Retrieve and print defect lists.
- disk** Choose the disk that will be used in subsequent operations (known as the current disk.)
- fdisk** (x86 systems only.) Run the **fdisk(1M)** program to create a Solaris **fdisk** partition.
- format** Format and verify the current disk.
- inquiry** Display the vendor, product name, and revision level of the current drive.
- label** Write a new label to the current disk.
- partition** Create and modify slices.
- quit** Exit the format menu.
- repair** Repair a specific block on the disk.
- save** Save new disk and slice information.
- type** Identify the manufacturer and model of the current disk.
- verify** Print the following information about the disk: manufacturer and model, number of cylinders, alternate cylinders, heads and sectors, and partition table.
- volname** Label the disk with a new eight character volume name.

ENVIRONMENT

- FORMAT_PATH** a colon-separated list of filenames and/or directories of disk and partition definitions. If a directory is specified, **format** searches for the file **format.dat** in that directory.

FILES	/etc/format.dat default data file
SEE ALSO	fmthard(1M) , prtvtoc(1M) , format.dat(4) , ipi(7D) , sd(7D) See <i>Disk Management</i> in <i>System Administration Guide, Volume I</i> .
x86 Only	fdisk(1M)
WARNINGS	When the format function is selected to format the Maxtor 207MB disk, the following message displays: Mode sense page(4) reports rpm value as 0, adjusting it to 3600 This is a drive bug that may also occur with older third party drives. The above message is not an error; the drive will still function correctly.
NOTES	format provides a help facility you can use whenever format is expecting input. You can request help about what information is expected by simply entering a question mark (?) and format prints a brief description of what type of input is needed. If you enter a ? at the menu prompt, a list of available commands is displayed. For SCSI disks, formatting is done with both Primary and Grown defects list by default. However, if only Primary list is extracted in defect menu before formatting, formatting will be done with Primary list only.

NAME	fsck – check and repair file systems
SYNOPSIS	<pre>fsck [-F <i>FSType</i>] [-m] [-V] [<i>special</i> ...] fsck [-F <i>FSType</i>] [-n N y Y] [-V] [-o <i>FSType-specific-options</i>] [<i>special</i> ...]</pre>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>fsck audits and interactively repairs inconsistent file system conditions. If the file system is inconsistent the default action for each correction is to wait for the user to respond yes or no. If the user does not have write permission fsck defaults to a no action. Some corrective actions will result in loss of data. The amount and severity of data loss may be determined from the diagnostic output.</p> <p><i>FSType-specific-options</i> are options specified in a comma-separated (with no intervening spaces) list of options or keyword-attribute pairs for interpretation by the <i>FSType</i>-specific module of the command.</p> <p><i>special</i> represents the block or character special device (for example, <code>/dev/rdisk/c1d0s7</code>) on which the file system resides. In general, the character special device should be used. fsck will not work on a block device if it is mounted.</p> <p>If no <i>special</i> device is specified fsck checks the file systems listed in <code>/etc/vfstab</code>. Those entries in <code>/etc/vfstab</code> which have a character special device entry in the fsckdev field and have a non-zero numeric entry in the fsckpass field will be checked. Specifying -F <i>FSType</i> limits the file systems to be checked to those of the type indicated.</p> <p>If <i>special</i> is specified, but -F is not, the file system type will be determined by looking for a matching entry in <code>/etc/vfstab</code>. If no entry is found, the default local file system type specified in <code>/etc/default/fs</code> will be used.</p> <p>If a file system type supports parallel checking (for example, ufs) some file systems eligible for checking may be checked in parallel. Consult the file system-specific man page (for example, fsck_ufs(1M)) for more information.</p>
OPTIONS	<p>-F <i>FSType</i> Specify the file system type on which to operate.</p> <p>-m Check but do not repair. This option checks that the file system is suitable for mounting, returning the appropriate exit status. If the file system is ready for mounting, fsck displays a message such as:</p> <p style="padding-left: 40px;">ufs fsck: sanity check: /dev/rdisk/c0t3d0s1 okay</p> <p>-n N Assume a no response to all questions asked by fsck; do not open the file system for writing.</p> <p>-V Echo the expanded command line but do not execute the command. This option may be used to verify and validate the command line.</p>

- y | Y** Assume a **yes** response to all questions asked by **fsck**.
- o *specific-options*** These *specific-options* can be any combination of the following separated by commas (with no intervening spaces).
- b=*n*** Use block *n* as the super block for the file system. Block 32 is always one of the alternate super blocks. The location of other super blocks may be determined by running **newfs(1M)** with the **-N*v*** options specified.
 - c** If the file system is in the old (static table) format, convert it to the new (dynamic table) format. If the file system is in the new format, convert it to the old format provided the old format can support the file system configuration. In interactive mode, **fsck** will list the direction the conversion is to be made and ask whether the conversion should be done. If a negative answer is given, no further operations are done on the file system. In preen mode, the direction of the conversion is listed and done if possible without user interaction. Conversion in preen mode is best used when all the file systems are being converted at once. The format of a file system can be determined from the first line of output from **fstyp(1M)**.
It should be noted that the **c** option is seldom used and is included only for compatibility with pre-4.1 releases. There is no guarantee that this option will be included in future releases.
 - f** Force checking of file systems regardless of the state of their super block clean flag.
 - p** (“preen”) Check and fix the file system non-interactively. Exit immediately if there is a problem requiring intervention. This option is required to enable parallel file system checking.
 - w** Check writable file systems only.

EXIT CODES

- 0** file system is okay and does not need checking
- 1** erroneous parameters are specified
- 32** file system is unmounted and needs checking (**fsck -m** only)
- 33** file system is already mounted
- 34** cannot stat device
- 36** uncorrectable errors detected - terminate normally
- 37** a signal was caught during processing
- 39** uncorrectable errors detected - terminate immediately
- 40** for root, same as **0**.

FILES	/etc/default/fs default local file system type. Default values can be set for the following flags in /etc/default/fs . For example: LOCAL=ufs LOCAL: The default partition for a command if no FSType is specified. /etc/vfstab list of default parameters for each file system
SEE ALSO	cli(1M) , crash(1M) , fsck_ufs(1M) , fsdb_ufs(1M) , fsirand(1M) , fstyp(1M) , mkfs(1M) , mkfs_ufs(1M) , mountall(1M) , newfs(1M) , reboot(1M) , fs_ufs(4) , vfstab(4) Manual pages for the <i>FSType</i> -specific modules of fsck .
WARNINGS	The operating system buffers file system data. Running fsck on a mounted file system can cause the operating system's buffers to become out of date with respect to the disk. For this reason, the file system should be <i>unmounted</i> when fsck is used. If this is not possible, care should be taken that the system is quiescent and that it is rebooted immediately after fsck is run. Quite often, however, this will not be sufficient. A panic will probably occur if running fsck on a file system modifies the file system.
NOTES	This command may not be supported for all <i>FSTypes</i> .

NAME	fsck_cachefs – check integrity of data cached with CacheFS
SYNOPSIS	fsck -F cachefs [-m -o noclean] <i>cache_directory</i>
DESCRIPTION	The CacheFS version of the fsck command checks the integrity of a cache directory. By default it corrects any CacheFS problems it finds. There is no interactive mode. The most likely invocation of fsck for CacheFS file systems is at boot time from an entry in the /etc/vfstab file (see vfstab(4)).
OPTIONS	Two command line options are available: -m Check, but do not repair. -o noclean Force a check on the cache even if there is no reason to suspect there is a problem.
EXAMPLES	The following example forces a check on the cache directory /cache3 : example% fsck -F cachefs -o noclean /cache3
SEE ALSO	cfsadmin(1M) , fsck(1M) , mount_cachefs(1M) , vfstab(4)

NAME	fsck_s5fs – file system consistency check and interactive repair
SYNOPSIS	fsck -F s5fs [<i>generic_options</i>] [<i>special ...</i>] fsck -F s5fs [<i>generic_options</i>] [-o <i>specific-options</i>] [<i>special ...</i>]
AVAILABILITY	x86 SUNWs53
DESCRIPTION	<p>fsck audits and interactively repairs inconsistent conditions on file systems. A file system to be checked may be specified by giving the name of the block or character special device or by giving the name of its mount point if a matching entry exists in /etc/vfstab. If no special device is specified, all s5 file systems specified in the vfstab with a fsckdev entry will be checked.</p> <p>In the case of correcting serious inconsistencies, by default, fsck asks for confirmation before making a repair and waits for the operator to respond either yes or no. If the operator does not have write permission on the file system, fsck will default to a -n (no corrections) action. See fsck(1M).</p> <p>Repairing some file system inconsistencies may result in loss of data. The amount and severity of data loss may be determined from the diagnostic output.</p> <p>fsck automatically corrects innocuous inconsistencies such as unreferenced inodes, missing blocks in the free list, blocks appearing in the free list and also in files, or incorrect counts in the superblock automatically. It displays a message for each inconsistency corrected that identifies the nature of the correction on which the file system took place. After successfully correcting a file system, fsck prints the number of files on that file system and the number of used and free blocks.</p> <p>Inconsistencies checked are as follows:</p> <ul style="list-style-type: none">• Blocks claimed by more than one inode or the free list.• Blocks claimed by an inode or the free list outside the range of the file system.• Incorrect link counts.• Incorrect directory sizes.• Bad inode format.• Blocks not accounted for anywhere.• Directory checks, file pointing to unallocated inode, inode number out of range, absence of '.' and '..' entries in any directory.• Superblock checks: more blocks for inodes than there are in the file system.• Bad free block list format.• Total free block and/or free inode count incorrect. <p>Orphaned files and directories (allocated but unreferenced) are, with the operator's concurrence, reconnected by placing them in the lost+found directory. The name assigned is the inode number. If the lost+found directory does not exist, it is created.</p>

OPTIONS

See generic **fsck(1M)** for *generic_options* and details for specifying *special*.

- o** Specify **s5** file system specific options. These options can be any combination of the following separated by commas (with no intervening spaces):
 - f** or **F** Fast check; duplicate blocks and free list check only.
 - I** After all other output is done, print i-number/pathname correspondences for damaged files.
 - t** *scratchfile* If there is insufficient memory and a temporary file is necessary to complete file system checking, use *scratchfile* as the temporary file.
 - T** *scratchfile* Same as above.
 - s** *cyl:skip* If it is necessary to rewrite (salvage) the free block list to correct an inconsistency, interleave the blocks such that, to the extent possible within each group of *cyl* consecutive free blocks, the interval between blocks is *skip*. For example, with an interleave of 8:3, in each group of eight consecutive free blocks, the order on the free list would be 1 4 7 2 5 8 3 6. If no *cyl:skip* is given, the value is either taken from the superblock, or, if unspecified (either has a value of 0), 400:7 is used. For obscure historical reasons, interleave specification of "3" and "4" (without colons) are taken to mean 200:5 and 418:7, respectively.
 - S** *cyl:skip* Same as above, except rewrite the free block list unconditionally.
 - q** Quiet; produce less verbose output.
 - D** Perform more extensive directory checking than normal.
 - p** ("preen") Check and fix the file system non-interactively. Exit immediately if there is a problem requiring intervention.
 - ?** Print usage message.

FILES /etc/vfstab list of default parameters for each file system

SEE ALSO **fsck(1M)**

NOTES It is usually faster to check the character special device than the block special device.

NAME	fsck_ufs – file system consistency check and interactive repair
SYNOPSIS	fsck -F ufs [<i>generic-options</i>] [<i>special</i> ...] fsck -F ufs [<i>generic-options</i>] [-o <i>specific-options</i>] [<i>special</i> ...]
DESCRIPTION	<p>fsck audits and interactively repairs inconsistent conditions on file systems. A file system to be checked may be specified by giving the name of the block or character <i>special</i> device or by giving the name of its mount point if a matching entry exists in /etc/vfstab.</p> <p><i>special</i> represents the block or character special device (for example, /dev/rdisk/c1d0s7) on which the file system resides. In general, the character special device should be used. fsck will not work on a block device if the block device is mounted, unless the file system is error-locked.</p> <p>If no <i>special</i> device is specified, all ufs file systems specified in the vfstab with a fsckdev entry will be checked. If the -p (“preen”) option is specified, ufs file systems with an fsckpass number greater than one are checked in parallel. See fsck(1M).</p> <p>In the case of correcting serious inconsistencies, by default, fsck asks for confirmation before making a repair and waits for the operator to respond either yes or no. If the operator does not have write permission on the file system, fsck will default to a -n (no corrections) action. See fsck(1M).</p> <p>Repairing some file system inconsistencies may result in loss of data. The amount and severity of data loss may be determined from the diagnostic output.</p> <p>fsck automatically corrects innocuous inconsistencies such as unreferenced inodes, too-large link counts in inodes, missing blocks in the free list, blocks appearing in the free list and also in files, or incorrect counts in the super block automatically. It displays a message for each inconsistency corrected that identifies the nature of the correction on which the file system took place. After successfully correcting a file system, fsck prints the number of files on that file system, the number of used and free blocks, and the percentage of fragmentation.</p> <p>Inconsistencies checked are as follows:</p> <ul style="list-style-type: none"> • Blocks claimed by more than one inode or the free list. • Blocks claimed by an inode or the free list outside the range of the file system. • Incorrect link counts. • Incorrect directory sizes. • Bad inode format. • Blocks not accounted for anywhere. • Directory checks, file pointing to unallocated inode, inode number out of range, absence of ‘.’ and ‘.’ as the first two entries in each directory. • Super Block checks: more blocks for inodes than there are in the file system. • Bad free block list format. • Total free block and/or free inode count incorrect.

Orphaned files and directories (allocated but unreferenced) are, with the operator's concurrence, reconnected by placing them in the **lost+found** directory. The name assigned is the inode number. If the **lost+found** directory does not exist, it is created. If there is insufficient space in the **lost+found** directory, its size is increased.

OPTIONS

The *generic-options* consist of the following options:

- m** Check but do not repair. This option checks that the file system is suitable for mounting, returning the appropriate exit status. If the file system is ready for mounting, **fsck** displays a message such as:
ufs fsck: sanity check: /dev/rdisk/c0t3d0s1 okay
- n | N** Assume a **no** response to all questions asked by **fsck**; do not open the file system for writing.
- V** Echo the expanded command line, but do not execute the command. This option may be used to verify and validate the command line.
- y | Y** Assume a **yes** response to all questions asked by **fsck**.

See generic **fsck**(1M) for the details for specifying *special*.

- o *specific-options*** Specify **ufs** file system specific options. These options can be any combination of the following separated by commas (with no intervening spaces).
 - b=n** Use block *n* as the super block for the file system. Block 32 is always one of the alternate super blocks. The location of other super blocks may be determined by running **newfs**(1M) with the **-Nv** options specified.
 - c** If the file system is in the old (static table) format, convert it to the new (dynamic table) format. If the file system is in the new format, convert it to the old format provided the old format can support the file system configuration. In interactive mode, **fsck** will list the direction the conversion is to be made and ask whether the conversion should be done. If a negative answer is given, no further operations are done on the file system. In preen mode, the direction of the conversion is listed and done if possible without user interaction. Conversion in preen mode is best used when all the file systems are being converted at once. The format of a file system can be determined from the first line of output from **fstyp**(1M).
 It should be noted that the **c** option is seldom used and is included only for compatibility with pre-4.1 releases. There is no guarantee that this option will be included in future releases.
 - f** Force checking of file systems regardless of the state of their super block clean flag.

- p** (“preen”) Check and fix the file system non-interactively. Exit immediately if there is a problem requiring intervention. This option is required to enable parallel file system checking.
- w** Check writable file systems only.

FILES /etc/vfstab list of default parameters for each file system

SEE ALSO [cli\(1M\)](#), [crash\(1M\)](#), [fsck\(1M\)](#), [fsdb_ufs\(1M\)](#), [fsirand\(1M\)](#), [fstyp\(1M\)](#), [mkfs\(1M\)](#), [mkfs_ufs\(1M\)](#), [mountall\(1M\)](#), [newfs\(1M\)](#), [reboot\(1M\)](#), [fs_ufs\(4\)](#), [vfstab\(4\)](#)

WARNINGS The operating system buffers file system data. Running **fsck** on a mounted file system can cause the operating system’s buffers to become out of date with respect to the disk. For this reason, the file system should be *unmounted* when **fsck** is used. If this is not possible, care should be taken that the system is quiescent and that it is rebooted immediately after **fsck** is run. Quite often, however, this will not be sufficient. A panic will probably occur if running **fsck** on a file system modifies the file system.

NOTES It is usually faster to check the character special device than the block special device.

NAME	fsdb – file system debugger
SYNOPSIS	fsdb [-F <i>FSType</i>] [-V] [-o <i>FSType-specific_options</i>] <i>special</i>
AVAILABILITY	SUNWcsu
DESCRIPTION	fsdb is a file system debugger that allows for the manual repair of a file system after a crash. <i>special</i> is a special device used to indicate the file system to be debugged. fsdb is intended for experienced users only. <i>FSType</i> is the file system type to be debugged. Since different <i>FSTypes</i> have different structures and hence different debugging capabilities, the manual pages for the <i>FSType</i> -specific fsdb should be consulted for a more detailed description of the debugging capabilities.
OPTIONS	<p>-F Specify the <i>FSType</i> on which to operate. The <i>FSType</i> should either be specified here or be determinable from /etc/vfstab by matching the <i>special</i> with an entry in the table, or by consulting /etc/default/fs.</p> <p>-V Echo the complete command line, but do not execute the command. The command line is generated by using the options and arguments provided by the user and adding to them information derived from /etc/vfstab. This option may be used to verify and validate the command line.</p> <p>-o Specify <i>FSType</i>-specific options.</p>
FILES	<p>/etc/default/fs default local file system type. Default values can be set for the following flags in /etc/default/fs. For example: LOCAL=ufs</p> <p>LOCAL: The default partition for a command if no <i>FSType</i> is specified.</p> <p>/etc/vfstab list of default parameters for each file system</p>
SEE ALSO	vfstab(4) Manual pages for the <i>FSType</i> -specific modules of fsdb .
NOTES	This command may not be supported for all <i>FSTypes</i> .

NAME	fsdb_ufs – ufs file system debugger								
SYNOPSIS	fsdb -F ufs [<i>generic_options</i>] [<i>specific_options</i>] <i>special</i>								
AVAILABILITY	SUNWxxx								
DESCRIPTION	<p>The fsdb_ufs command is an interactive tool that can be used to patch up a damaged UFS file system. It has conversions to translate block and i-numbers into their corresponding disk addresses. Also included are mnemonic offsets to access different parts of an inode. These greatly simplify the process of correcting control block entries or descending the file system tree.</p> <p>fsdb contains several error-checking routines to verify inode and block addresses. These can be disabled if necessary by invoking fsdb with the -o option or by the use of the o command.</p> <p>fsdb reads a block at a time and will therefore work with raw as well as block I/O devices. A buffer management routine is used to retain commonly used blocks of data in order to reduce the number of read system calls. All assignment operations result in an immediate write-through of the corresponding block. Note that in order to modify any portion of the disk, fsdb must be invoked with the w option.</p> <p>Wherever possible, adb-like syntax was adopted to promote the use of fsdb through familiarity.</p>								
OPTIONS	<p>The following option is supported:</p> <p>-o Specify UFS file system specific options. These options can be any combination of the following separated by commas (with no intervening spaces). The options available are:</p> <table border="0" style="margin-left: 2em;"> <tr> <td>?</td> <td>Display usage</td> </tr> <tr> <td>o</td> <td>Override some error conditions</td> </tr> <tr> <td>p='string'</td> <td>set prompt to string</td> </tr> <tr> <td>w</td> <td>open for write</td> </tr> </table>	?	Display usage	o	Override some error conditions	p='string'	set prompt to string	w	open for write
?	Display usage								
o	Override some error conditions								
p='string'	set prompt to string								
w	open for write								
USAGE	<p>Numbers are considered hexadecimal by default. However, the user has control over how data is to be displayed or accepted. The base command will display or set the input/output base. Once set, all input will default to this base and all output will be shown in this base. The base can be overridden temporarily for input by preceding hexadecimal numbers with '0x', preceding decimal numbers with '0t', or octal numbers with '0'. Hexadecimal numbers beginning with a-f or A-F must be preceded with '0x' to distinguish them from commands.</p> <p>Disk addressing by fsdb is at the byte level. However, fsdb offers many commands to convert a desired inode, directory entry, block, superblock etc. to a byte address. Once the address has been calculated, fsdb will record the result in dot (see next paragraph).</p>								

Several global values are maintained by **fsdb**:

- the current base (referred to as **base**),
- the current address (referred to as **dot**),
- the current inode (referred to as **inode**),
- the current count (referred to as **count**),
- and the current type (referred to as **type**).

Most commands use the preset value of **dot** in their execution. For example,

```
> 2:inode
```

will first set the value of **dot** to 2, ':' will alert the start of a command, and the **inode** command will set **inode** to 2. A count is specified after a ','. Once set, **count** will remain at this value until a new command is encountered which will then reset the value back to 1 (the default). So, if

```
> 2000,400/X
```

is typed, 400 hex longs are listed from 2000, and when completed, the value of **dot** will be $2000 + 400 * \text{sizeof}(\text{long})$. If a carriage-return is then typed, the output routine will use the current values of **dot**, **count**, and **type** and display 400 more hex longs. A '*' will cause the entire block to be displayed.

End of fragment, block and file are maintained by **fsdb**. When displaying data as fragments or blocks, an error message will be displayed when the end of fragment or block is reached. When displaying data using the **db**, **ib**, **directory**, or **file** commands an error message is displayed if the end of file is reached. This is mainly needed to avoid passing the end of a directory or file and getting unknown and unwanted results.

An example showing several commands and the use of carriage-return would be:

```
> 2:ino; 0:dir?d
```

or

```
> 2:ino; 0:db:block?d
```

The two examples are synonymous for getting to the first directory entry of the root of the file system. Once there, subsequent carriage-returns (or +, -) will advance to subsequent entries. Note that

```
> 2:inode; :ls
```

or

```
> :ls /
```

is again synonymous.

Expressions

The symbols recognized by **fsdb** are:

CARRIAGE-RETURN

update the value of **dot** by the current value of **type** and display using the current value of **count**.

#

numeric expressions may be composed of +, -, *, and % operators (evaluated left to right) and may use parentheses. Once evaluated, the value of **dot** is updated.

, <i>count</i>	count indicator. The global value of count will be updated to count . The value of count will remain until a new command is run. A count specifier of '*' will attempt to show a <i>blocks</i> 's worth of information. The default for count is 1.
? <i>f</i>	display in structured style with format specifier <i>f</i> (see Formatted Output section).
/ <i>f</i>	display in unstructured style with format specifier <i>f</i> (see Formatted Output section).
.	the value of dot .
+ <i>e</i>	increment the value of dot by the expression <i>e</i> . The amount actually incremented is dependent on the size of type : $\mathbf{dot} = \mathbf{dot} + e * \mathbf{sizeof}(\mathbf{type})$ The default for <i>e</i> is 1.
- <i>e</i>	decrement the value of dot by the expression <i>e</i> (see +).
* <i>e</i>	multiply the value of dot by the expression <i>e</i> . Multiplication and division don't use type . In the above calculation of dot , consider the sizeof(type) to be 1.
% <i>e</i>	divide the value of dot by the expression <i>e</i> (see *).
< <i>name</i>	restore an address saved in register <i>name</i> . <i>name</i> must be a single letter or digit.
> <i>name</i>	save an address in register <i>name</i> . <i>name</i> must be a single letter or digit.
= <i>f</i>	display indicator. If <i>f</i> is a legitimate format specifier (see Formatted Output section), then the value of dot is displayed using format specifier <i>f</i> . Otherwise, assignment is assumed (see next item).
= [<i>s</i>] [<i>e</i>]	assignment indicator. The address pointed to by dot has its contents changed to the value of the expression <i>e</i> or to the ASCII representation of the quoted (") string <i>s</i> . This may be useful for changing directory names or ASCII file information.
+= <i>e</i>	incremental assignment. The address pointed to by dot has its contents incremented by expression <i>e</i> .
-= <i>e</i>	decremental assignment. The address pointed to by dot has its contents decremented by expression <i>e</i> .

Commands

A command must be prefixed by a ':' character. Only enough letters of the command to uniquely distinguish it are needed. Multiple commands may be entered on one line by separating them by a space, tab or ';'.

In order to view a potentially unmounted disk in a reasonable manner, **fsdb** offers the *cd*, *pwd*, *ls* and *find* commands. The functionality of these commands substantially matches those of its UNIX counterparts (see individual command for details). The '*', '?', and '['-] wild card characters are available.

base=b	display or set base. As stated above, all input and output is governed by the current base . If the '=b' is left off, the current base is displayed. Otherwise, the current base is set to <i>b</i> . Note that this is interpreted using the old value of base , so to ensure correctness use the '0', '0t', or '0x' prefix when changing the base . The default for base is hexadecimal.
block	convert the value of dot to a block address.
cd dir	change the current directory to directory <i>dir</i> . The current values of inode and dot are also updated. If no <i>dir</i> is specified, then change directories to inode 2 ("/").
cg	convert the value of dot to a cylinder group.
directory	If the current inode is a directory, then the value of dot is converted to a directory slot offset in that directory and dot now points to this entry.
file	the value of dot is taken as a relative block count from the beginning of the file. The value of dot is updated to the first byte of this block.
find dir [-name n] [-inum i]	find files by name or i-number. find recursively searches directory dir and below for filenames whose i-number matches <i>i</i> or whose name matches pattern <i>n</i> . Note that only one of the two options (-name or -inum) may be used at one time. Also, the -print is not needed or accepted.
fill=p	fill an area of disk with pattern <i>p</i> . The area of disk is delimited by dot and count .
fragment	convert the value of <i>dot</i> to a fragment address. The only difference between the fragment command and the block command is the amount that is able to be displayed.
inode	convert the value of <i>dot</i> to an inode address. If successful, the current value of inode will be updated as well as the value of <i>dot</i> . As a convenient shorthand, if ':inode' appears at the beginning of the line, the value of <i>dot</i> is set to the current inode and that inode is displayed in inode format.
ls [-R] [-l] pat1 pat2 ...	list directories or files. If no file is specified, the current directory is assumed. Either or both of the options may be used (but, if used, <i>must</i> be specified before the filename specifiers). Also, as stated above, wild card characters are available and multiple arguments may be given. The long listing shows only the i-number and the name; use the inode command with '?i' to get more information.
override	toggle the value of override. Some error conditions may be overridden if override is toggled on.
prompt p	change the fsdb prompt to <i>p</i> . <i>p</i> must be surrounded by (")s.
pwd	display the current working directory.

quit	quit fsdb .
sb	the value of <i>dot</i> is taken as a cylinder group number and then converted to the address of the superblock in that cylinder group. As a shorthand, 'sb' at the beginning of a line will set the value of <i>dot</i> to <i>the</i> superblock and display it in superblock format.
!	escape to shell

Inode Commands

In addition to the above commands, there are several commands that deal with inode fields and operate directly on the current **inode** (they still require the **'.'**). They may be used to more easily display or change the particular fields. The value of *dot* is only used by the **':db'** and **':ib'** commands. Upon completion of the command, the value of *dot* is changed to point to that particular field. For example,

```
> :ln=+1
```

would increment the link count of the current **inode** and set the value of *dot* to the address of the link count field.

at	access time.
bs	block size.
ct	creation time.
db	use the current value of <i>dot</i> as a direct block index, where direct blocks number from 0 - 11. In order to display the block itself, you need to 'pipe' this result into the block or fragment command. For example, <pre>> 1:db:block,20/X</pre> would get the contents of data block field 1 from the inode and convert it to a block address. 20 longs are then displayed in hexadecimal (see Formatted Output sub-section).
gid	group id.
ib	use the current value of <i>dot</i> as an indirect block index where indirect blocks number from 0 - 2. This will only get the indirect block itself (the block containing the pointers to the actual blocks). Use the file command and start at block 12 to get to the actual blocks.
ln	link count.
mt	modification time.
md	mode.
maj	major device number.
min	minor device number.

nm although listed here, this command actually operates on the directory name field. Once poised at the desired directory entry (using the *directory* command), this command will allow you to change or display the directory name. For example,

```
> 7:dir:nm="foo"
```

will get the 7th directory entry of the current **inode** and change its name to foo. Note that names cannot be made larger than the field is set up for. If an attempt is made, the string is truncated to fit and a warning message to this effect is displayed.

sz file size.

uid user id.

Formatted Output

There are two styles and many format types. The two styles are structured and unstructured. Structured output is used to display inodes, directories, superblocks and the like. Unstructured just displays raw data. The following table shows the different ways of displaying:

```
?
  c      display as cylinder groups
  i      display as inodes
  d      display as directories
  s      display as superblocks

/
  b      display as bytes
  c      display as characters
  o O    display as octal shorts or longs
  d D    display as decimal shorts or longs
  x X    display as hexadecimal shorts or longs
```

The format specifier immediately follows the '/' or '?' character. The values displayed by '/b' and all '?' formats are displayed in the current **base**. Also, **type** is appropriately updated upon completion.

EXAMPLES

```
> 2000+400%(20+20)=D
    will display 2010 in decimal (use of fsdb as a calculator for complex arithmetic).

> 386:ino?i
    display i-number 386 in an inode format. This now becomes the current inode.

> :ln=4
    changes the link count for the current inode to 4.

> :ln=+1
    increments the link count by 1.

> :ct=X
    display the creation time as a hexadecimal long.

> :mt=t
    display the modification time in time format.

> 0:file/c
    displays, in ASCII, block zero of the file associated with the current inode.
```

- > **2:ino,*?d** displays the first blocks worth of directory entries for the root inode of this file system. It will stop prematurely if the EOF is reached.
- > **5:dir:inode; 0:file,*?c** changes the current inode to that associated with the 5th directory entry (numbered from zero) of the current **inode**. The first logical block of the file is then displayed in ASCII.
- > **:sb** displays the superblock of this file system.
- > **1:cg?c** displays cylinder group information and summary for cylinder group 1.
- > **2:inode; 7:dir=3** changes the i-number for the seventh directory slot in the root directory to 3.
- > **7:dir:nm="name"** changes the name field in the directory slot to *name*.
- > **2:db:block,*?d** displays the third block of the current **inode** as directory entries.
- > **3c3:fragment,20:fill=0x20** get fragment **3c3** and fill **20 type** elements with **0x20**.
- > **2050=0xffff** set the contents of address **2050** to **0xffffffff**. **0xffffffff** may be truncated depending on the current **type**.
- > **1c92434="this is some text"** will place the ASCII for the string at **1c92434**.

SEE ALSO [crl\(1M\)](#), [fsck_ufs\(1M\)](#), [dir_ufs\(4\)](#), [fs_ufs\(4\)](#)

WARNINGS Since **fsdb** reads the disk raw, extreme caution is advised in determining its availability of **fsdb** on the system. Suggested permissions are 600 and owned by bin.

NOTES The old command line syntax for clearing i-nodes using the ufs-specific '**-z i-number**' option is still supported by the new debugger, though it is obsolete and will be removed in a future release. Use of this flag will result in correct operation, but an error message will be printed warning of the impending obsolescence of this option to the command. The equivalent functionality is available using the more flexible [crl\(1M\)](#) command.

NAME	fsirand – install random inode generation numbers
SYNOPSIS	fsirand [-p] <i>special</i>
DESCRIPTION	<p>fsirand installs random inode generation numbers on all the inodes on device <i>special</i>, and also installs a file system ID in the superblock. This helps increase the security of file systems exported by NFS.</p> <p>fsirand must be used only on an unmounted file system that has been checked with fsck(1M). The only exception is that it can be used on the root file system in single-user mode, if the system is immediately re-booted afterwards.</p>
OPTIONS	-p Print out the generation numbers for all the inodes, but do not change the generation numbers.
SEE ALSO	fsck(1M)

NAME	fstyp – determine file system type
SYNOPSIS	fstyp [-v] <i>special</i>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>fstyp allows the user to determine the file system type of unmounted file systems using heuristic programs.</p> <p>An fstyp module for each file system type to be checked is executed; each of these modules applies an appropriate heuristic to determine whether the supplied <i>special</i> file is of the type for which it checks. If it is, the program prints on standard output the usual file system identifier for that type (for example, “ufs”) and exits with a return code of 0; if none of the modules succeed, the error message unknown_fstyp (no matches) is returned and the exit status is 1. If more than one module succeeds, the error message unknown_fstyp (multiple matches) is returned and the exit status is 2.</p>
OPTIONS	-v Produce verbose output. This is usually information about the file systems superblock and varies across different <i>FSTypes</i> .
NOTES	The use of heuristics implies that the result of fstyp is not guaranteed to be accurate.

NAME	fusage – disk access profiler				
SYNOPSIS	/usr/sbin/fusage [<i>mount_point</i> <i>block_special_device</i> ...]				
AVAILABILITY	SUNWcsu				
DESCRIPTION	<p>When used with no options, fusage reports block I/O transfers, in kilobytes, to and from all locally mounted file systems. The count data are cumulative since the time of the mount.</p> <p>The report includes one section for each file system and advertised resource, and has one entry for each remotely mounted file system. Sections are ordered by device name; advertised resources that are not complete file systems will immediately follow the sections for the file systems they are in.</p>				
OPTIONS	<table><tr><td><i>mount_point</i></td><td>Reports on the named file system.</td></tr><tr><td><i>block_special_device</i></td><td>Block special device.</td></tr></table>	<i>mount_point</i>	Reports on the named file system.	<i>block_special_device</i>	Block special device.
<i>mount_point</i>	Reports on the named file system.				
<i>block_special_device</i>	Block special device.				
SEE ALSO	crash(1M) , df(1M) , mount(1M)				

NAME	fuser – identify processes using a file or file structure
SYNOPSIS	/usr/sbin/fuser [-c f] ku files [-c f] ku files ...
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>fuser displays the process IDs of the processes that are using the <i>files</i> specified as arguments. Each process ID is followed by a letter code, interpreted as follows: if the process is using the file as 1) its current directory, the code is c; 2) its root directory, the code is r; 3) an open file, the code is o; or 4) its text file, the code is t. For block special devices with mounted file systems, all processes using any file on that device are listed. For all types of files (text files, executables, directories, devices, and so on), only the processes using that file are reported.</p> <p>If more than one group of files are specified, the options may be respecified for each additional group of files. A lone dash cancels the options currently in force.</p> <p>The process IDs are printed as a single line on the standard output, separated by spaces and terminated with a single new line. All other output is written on standard error.</p> <p>Any user with permission to read /dev/kmem and /dev/mem can use fuser. Only the super-user can terminate another user's process.</p>
OPTIONS	<p>-c Report on files that are mount points for file systems, and any files within that mounted file system.</p> <p>-f Print a report for the named file, not for files within a mounted file system.</p> <p>-k Send the SIGKILL signal to each process. Since this option spawns kills for each process, the kill messages may not show up immediately (see kill(2)).</p> <p>-u Display the user login name in parentheses following the process ID.</p>
SEE ALSO	ps(1) , mount(1M) , kill(2) , signal(3C)
NOTES	Because fuser works with a snapshot of the system image, it may miss processes that begin using a file while fuser is running. Also, processes reported as using a file may have stopped using it while fuser was running. These factors should discourage the use of the -k option.

NAME	fwtmp, wtmpfix – manipulate connect accounting records
SYNOPSIS	<code>/usr/lib/acct/fwtmp [-ic]</code> <code>/usr/lib/acct/wtmpfix [file...]</code>
DESCRIPTION	<p>fwtmp reads from the standard input and writes to the standard output, converting binary records of the type found in <code>/var/adm/wtmp</code> to formatted ASCII records. The ASCII version is useful when it is necessary to edit bad records.</p> <p>wtmpfix examines the standard input or named files in utmp.h format, corrects the time/date stamps to make the entries consistent, and writes to the standard output. A hyphen (-) can be used in place of <i>file</i> to indicate the standard input. If time/date corrections are not performed, acctcon(1M) will fault when it encounters certain date-change records.</p> <p>Each time the date is set, a pair of date change records are written to <code>/var/adm/wtmp</code>. The first record is the old date denoted by the string "old time" placed in the line field and the flag OLD_TIME placed in the type field of the utmp structure. The second record specifies the new date and is denoted by the string new time placed in the line field and the flag NEW_TIME placed in the type field. wtmpfix uses these records to synchronize all time stamps in the file.</p> <p>In addition to correcting time/date stamps, wtmpfix will check the validity of the name field to ensure that it consists solely of alphanumeric characters or spaces. If it encounters a name that is considered invalid, it will change the login name to INVALID and write a diagnostic to the standard error. In this way, wtmpfix reduces the chance that acctcon will fail when processing connect accounting records.</p>
OPTIONS	-ic Denotes that input is in ASCII form, and output is to be written in binary form.
FILES	<code>/var/adm/wtmp</code> <code>/usr/include/utmp.h</code>
SEE ALSO	acctcom (1), ed (1), acct (1M), acctcms (1M), acctcon (1M), acctmerg (1M), acctprc (1M), acctsh (1M), runacct (1M), acct (2), acct (4), utmp (4) <i>System Administration Guide, Volume II</i>

NAME	gencc – create a front-end to the cc command
SYNOPSIS	gencc
DESCRIPTION	<p>The gencc command is an interactive command designed to aid in the creation of a front-end to the cc command. Since hard-coded pathnames have been eliminated from the C Compilation System (CCS), it is possible to move pieces of the CCS to new locations without recompilation. The new locations of moved pieces can be specified through the -Y option to the cc command. However, it is inconvenient to supply the proper -Y options with every invocation of the cc command. Further, if a system administrator moves pieces of the CCS, such movement should be invisible to users.</p> <p>The front-end to the cc command that gencc generates is a one-line shell script that calls the cc command with the proper -Y options specified. The front-end to the cc command will also pass all user-supplied options to the cc command.</p> <p>gencc prompts for the location of each tool and directory that can be respecified by a -Y option to the cc command. If no location is specified, it assumes that that piece of the CCS has not been relocated. After all the locations have been prompted for, gencc will create the front-end to the cc command.</p> <p>gencc creates the front-end to the cc command in the current working directory and gives the file the same name as the cc command. Thus, gencc can not be run in the same directory containing the actual cc command. Further, if a system administrator has redistributed the CCS, the actual cc command should be placed in a location that is not typically in a user's path (for example, /usr/lib). Such placement will prevent users from accidentally invoking the cc command without using the front-end.</p>
FILES	./cc front-end to cc
SEE ALSO	cc(1B)
NOTES	<p>gencc does not produce any warnings if a tool or directory does not exist at the specified location. Also, gencc does not actually move any files to new locations.</p> <p>The gencc command is obsolete.</p>

NAME	getdev – lists devices based on criteria
SYNOPSIS	<code>/usr/bin/getdev [-ae] [criteria ...] [device ...]</code>
DESCRIPTION	<p>getdev generates a list of devices that match certain criteria. The criteria includes a list of attributes (given in expressions) and a list of devices. If no criteria are given, all devices are included in the list.</p> <p>Devices must satisfy at least one of the criteria in the list unless the -a option is used. Then, only those devices which match all of the criteria in a list will be included.</p> <p>Devices which are defined on the command line and which match the criteria are included in the generated list. However, if the -e option is used, the list becomes a set of devices to be <i>excluded</i> from the list.</p> <p><i>criteria</i> defines the criteria that a device must match to be included in the generated list. Should be specified by giving expressions.</p> <p><i>device</i> defines the devices which should be included in the generated list. This can be the pathname of the device or the device alias.</p>
Criteria Expression Types	<p>There are four possible expression types which the criteria specified in the <i>criteria</i> argument may follow:</p> <p><i>attribute=value</i> Selects all devices whose attribute <i>attribute</i> is defined and is equal to <i>value</i>.</p> <p><i>attribute!=value</i> Selects all devices whose attribute <i>attribute</i> is defined and does not equal <i>value</i>.</p> <p><i>attribute:*</i> Selects all devices which have the attribute <i>attribute</i> defined.</p> <p><i>attribute!:*</i> Selects all devices which do not have the attribute <i>attribute</i> defined.</p> <p>See the putdev(1M) manual page for a complete listing and description of available attributes.</p>
OPTIONS	<p>-a Specifies that a device must match all criteria to be included in the list generated by this command. The option has no effect if no criteria are defined.</p> <p>-e Specifies that the list of devices which follows on the command line should be <i>excluded</i> from the list generated by this command. (Without the -e the named devices are <i>included</i> in the generated list.) The flag has no effect if no devices are defined.</p>
ERRORS	<p>The command will exit with one of the following values:</p> <p>0 = Successful completion of the task.</p> <p>1 = Command syntax incorrect, invalid option used, or internal error occurred.</p> <p>2 = Device table could not be opened for reading.</p>

FILES

/etc/device.tab

SEE ALSO

devattr(1M), getdgrp(1M), putdev(1M), putdgrp(1M)

NAME	getdgrp – lists device groups which contain devices that match criteria
SYNOPSIS	<code>/usr/sbin/getdgrp [-ael] [criteria ...] [dgroup ...]</code>
DESCRIPTION	<p>getdgrp generates a list of device groups that contain devices matching the given criteria. The criteria is given in the form of expressions.</p> <p><i>criteria</i> can be one expression or a list of expressions which a device must meet for its group to be included in the list generated by getdgrp. If no criteria are given, all device groups are included in the list.</p> <p>Devices must satisfy at least one of the criteria in the list. However, the -a option can be used to define that a "logical and" operation should be performed. Then, only those groups containing devices which match all of the criteria in a list will be included.</p> <p><i>dgroup</i> defines a set of device groups to be included in the list. Device groups that are defined and which contain devices matching the criteria are included. However, if the -e option is used, this list defines a set of device groups to be excluded. When the -e option is used and <i>criteria</i> is also defined, the generated list will include device groups containing devices which match the criteria and are not in the command line list.</p> <p><i>criteria</i> defines criteria that a device must match before a device group to which it belongs can be included in the generated list.</p> <p><i>dgroup</i> defines device groups which should be included in or excluded from the generated list.</p>
Criteria Expression Types	<p>There are four possible expressions types which the criteria specified in the <i>criteria</i> argument may follow:</p> <p><i>attribute=value</i> Selects all device groups with a member whose attribute <i>attribute</i> is defined and is equal to <i>value</i>.</p> <p><i>attribute!=value</i> Selects all device groups with a member whose attribute <i>attribute</i> is defined and does not equal <i>value</i>.</p> <p><i>attribute:*</i> Selects all device groups with a member which has the attribute <i>attribute</i> defined.</p> <p><i>attribute!:*</i> Selects all device groups with a member which does not have the attribute <i>attribute</i> defined.</p> <p>See putdev(1M) for a complete listing and description of available attributes.</p>
OPTIONS	<p>-a Specifies that a device must match all criteria to be included in the list generated by this command. The option has no effect if no criteria are defined.</p> <p>-e Specifies that the list of device groups on the command line should be <i>excluded</i> from the list generated by this command. (Without the -e option the named device groups are <i>included</i> in the generated list.) The flag has no effect if no devices are defined.</p>

-l Specifies that all device groups (subject to the **-e** option and the *dgroup* list) should be listed even if they contain no valid device members. This option has no affect if *criteria* is specified on the command line.

ERRORS The command will exit with one of the following values:

- 0** Successful completion of the task.
- 1** Command syntax incorrect, invalid option used, or internal error occurred.
- 2** Device table or device group table could not be opened for reading.

FILES */etc/device.tab*
/etc/dgroup.tab

SEE ALSO *devattr(1M)*, *getdev(1M)*, *putdev(1M)*, *putdgrp(1M)*

NAME	getent – get entries from administrative database																
SYNOPSIS	<code>/usr/bin/getent database [key ...]</code>																
DESCRIPTION	<p>getent gets a list of entries from the administrative database specified by <i>database</i>. The information comes from one or more of the sources that are specified for the <i>database</i> in <code>/etc/nsswitch.conf</code>.</p> <p><i>database</i> is the name of the database to be examined. This can be passwd, group, hosts, services, protocols, ethers, or networks. For each of these databases, it uses the appropriate library routines described in getpwnam(3C), getgrnam(3C), gethostbyname(3N), getservbyname(3N), getprotobyname(3N), ethers(3N), and getnetbyname(3N), respectively.</p> <p>Each <i>key</i> must be in a format appropriate for searching on the respective database; for example, it can be a <i>username</i> or <i>numeric-uid</i> for passwd, <i>hostname</i> or <i>IP address</i> for hosts, or <i>service</i>, <i>service/protocol</i>, <i>port</i>, or <i>port/proto</i> for services.</p> <p>getent prints out the database entries that match each of the supplied keys, one per line, in the format of the matching administrative file: passwd(4), group(4), hosts(4), services(4), protocols(4), ethers(4), or networks(4). If no key is given, all entries returned by the corresponding enumeration library routine (for example, getpwent or gethostent) are printed.</p>																
ERRORS	<p>getent exits with one of the following values:</p> <p>0 Successful completion of the task.</p> <p>1 Command syntax incorrect, invalid option used, or internal error occurred.</p> <p>2 At least one of the specified entry names was not found in the database.</p> <p>3 There is no support for enumeration on this database.</p>																
FILES	<table border="0"> <tr> <td><code>/etc/nsswitch.conf</code></td> <td>name service switch configuration file</td> </tr> <tr> <td><code>/etc/passwd</code></td> <td>password file</td> </tr> <tr> <td><code>/etc/group</code></td> <td>group file</td> </tr> <tr> <td><code>/etc/hosts</code></td> <td>host name database</td> </tr> <tr> <td><code>/etc/services</code></td> <td>Internet services and aliases</td> </tr> <tr> <td><code>/etc/protocols</code></td> <td>protocol name database</td> </tr> <tr> <td><code>/etc/ethers</code></td> <td>Ethernet address to hostname database or domain</td> </tr> <tr> <td><code>/etc/networks</code></td> <td>network name database</td> </tr> </table>	<code>/etc/nsswitch.conf</code>	name service switch configuration file	<code>/etc/passwd</code>	password file	<code>/etc/group</code>	group file	<code>/etc/hosts</code>	host name database	<code>/etc/services</code>	Internet services and aliases	<code>/etc/protocols</code>	protocol name database	<code>/etc/ethers</code>	Ethernet address to hostname database or domain	<code>/etc/networks</code>	network name database
<code>/etc/nsswitch.conf</code>	name service switch configuration file																
<code>/etc/passwd</code>	password file																
<code>/etc/group</code>	group file																
<code>/etc/hosts</code>	host name database																
<code>/etc/services</code>	Internet services and aliases																
<code>/etc/protocols</code>	protocol name database																
<code>/etc/ethers</code>	Ethernet address to hostname database or domain																
<code>/etc/networks</code>	network name database																
SEE ALSO	ethers (3N), getgrnam (3C), gethostbyname (3N), getnetbyname (3N), getservbyname (3N), getprotobyname (3N), getpwnam (3C), nsswitch.conf (4)																

NAME	gettable – get DoD Internet format host table from a host
SYNOPSIS	<code>/usr/sbin/gettable host</code>
AVAILABILITY	SUNWnisu
DESCRIPTION	<p>gettable is a simple program used to obtain the DoD Internet host table from a “host-name” server. The specified <i>host</i> is queried for the table. The table is placed in the file hosts.txt.</p> <p>gettable operates by opening a TCP connection to the port indicated in the service specification for “hostname”. A request is then made for all names and the resultant information is placed in the output file.</p> <p>gettable is best used in conjunction with the htable(1M) program which converts the DoD Internet host table format to that used by the network library lookup routines.</p>
SEE ALSO	<p>htable(1M) Harrenstien, Ken, Mary Stahl, and Elizabeth Feinler, <i>HOSTNAME Server</i>, RFC 953, Network Information Center, SRI International, Menlo Park, Calif., October 1985.</p>
NOTES	Should allow requests for only part of the database.

NAME	getty – set terminal type, modes, speed, and line discipline	
SYNOPSIS	/usr/lib/saf/ttymon [-h] [-t <i>timeout</i>] <i>line</i> [<i>speed</i> [<i>type</i> [<i>linedisc</i>]]] /usr/lib/saf/ttymon -c <i>file</i>	
AVAILABILITY	SUNWcsr	
DESCRIPTION	<p>getty is a symbolic link to /usr/lib/saf/ttymon. It is included for compatibility with previous releases for the few applications that still call getty directly. getty can only be executed by the super-user, (a process with the user ID root). Initially getty prints the login prompt, waits for the user's login name, and then invokes the login command. getty attempts to adapt the system to the terminal speed by using the options and arguments specified on the command line.</p> <p>When given no optional arguments, getty specifies the following: The <i>speed</i> of the interface is set to 300 baud, either parity is allowed, new-line characters are converted to carriage return-line feed, and tab expansion is performed on the standard output. getty types the login prompt before reading the user's name a character at a time. If a null character (or framing error) is received, it is assumed to be the result of the user pressing the BREAK key. This will cause getty to attempt the next <i>speed</i> in the series. The series that getty tries is determined by what it finds in /etc/ttydefs.</p>	
OPTIONS	<i>line</i>	The name of a TTY line in /dev to which getty is to attach itself. getty uses this string as the name of a file in the /dev directory to open for reading and writing.
	-h	If the -h flag is not set, a hangup will be forced by setting the speed to zero before setting the speed to the default or a specified speed.
	-t <i>timeout</i>	Specifies that getty should exit if the open on the line succeeds and no one types anything in <i>timeout</i> seconds.
	<i>speed</i>	The <i>speed</i> argument is a label to a speed and TTY definition in the file /etc/ttydefs . This definition tells getty at what speed to run initially, what the initial TTY settings are, and what speed to try next, (should the user press the BREAK key to indicate that the speed is inappropriate). The default <i>speed</i> is 300 baud.
	<i>type</i> and <i>linedisc</i>	These options are obsolete and will be ignored.
	-c <i>file</i>	The -c option is no longer supported. Instead use /usr/sbin/sttydefs -l to list the contents of the /etc/ttydefs file and perform a validity check on the file.

FILES /etc/ttydefs

SEE ALSO ct(1C), login(1), sttydefs(1M), ttymon(1M), ioctl(2), tty(7D)

NAME	getvol – verifies device accessibility
SYNOPSIS	<i>/usr/bin/getvol -n [-l label] device</i> <i>/usr/bin/getvol [-f -F] [-ow] [-l label -x label] device</i>
DESCRIPTION	getvol verifies that the specified device is accessible and that a volume of the appropriate medium has been inserted. The command is interactive and displays instructional prompts, describes errors, and shows required label information. <i>device</i> names the device which should be verified for accessibility.
OPTIONS	<p>-n Runs the command in non-interactive mode. The volume is assumed to be inserted upon command invocation.</p> <p>-l label Specifies that the label <i>label</i> must exist on the inserted volume (can be overridden by the -o option).</p> <p>-f Formats the volume after insertion, using the format command defined for this device in the device table.</p> <p>-F Formats the volume after insertion and places a file system on the device. Also uses the format command defined for this device in the device table.</p> <p>-o Allows the administrator to override a label check.</p> <p>-w Allows administrator to write a new label on the device. User is prompted to supply the label text. This option is ineffective if the -n option is enabled.</p> <p>-x label Specifies that the label <i>label</i> must exist on the device. This option should be used in place of the -l option when the label can only be verified by visual means. Use of the option causes a message to be displayed asking the administrator to visually verify that the label is indeed <i>label</i>.</p>
ERRORS	The command will exit with one of the following values: <p>0 Successful completion of the task.</p> <p>1 Command syntax incorrect, invalid option used, or internal error occurred.</p> <p>3 Device table could not be opened for reading.</p>
FILES	<i>/etc/device.tab</i>
NOTES	This command uses the device table to determine the characteristics of the device when performing the volume label checking.

NAME	groupadd – add (create) a new group definition on the system
SYNOPSIS	<code>/usr/sbin/groupadd [-g gid [-o]] group</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	The groupadd command creates a new group definition on the system by adding the appropriate entry to the <code>/etc/group</code> file.
OPTIONS	<p>-g gid The group id for the new group. This group id must be a non-negative decimal integer below MAXUID as defined in the <code><param.h></code> header. The group ID defaults to the next available (unique) number above the highest number currently assigned. For example, if groups 100, 105, and 200 are assigned as groups, the next default group number will be 201. (Group IDs from 0-99 are reserved by SunOS for future applications.)</p> <p>-o This option allows the <i>gid</i> to be duplicated (non-unique).</p> <p><i>group</i> A string of printable characters that specifies the name of the new group, up to a maximum of eight characters. It may not include a colon (:) or newline (\n).</p>
ERRORS	<p>The groupadd command exits with one of the following values:</p> <p>0 Success.</p> <p>2 Invalid command syntax. A usage message for the groupadd command is displayed.</p> <p>3 An invalid argument was provided to an option.</p> <p>4 <i>gid</i> is not unique (when -o option is not used).</p> <p>9 <i>group</i> is not unique.</p> <p>10 Cannot update the <code>/etc/group</code> file.</p>
FILES	<code>/etc/group</code>
SEE ALSO	users(1B) , groupdel(1M) , groupmod(1M) , logins(1M) , useradd(1M) , userdel(1M) , usermod(1M) , group(4)
NOTES	groupadd only adds a group definition to the local system. If a network nameservice such as NIS or NIS+ is being used to supplement the local <code>/etc/group</code> file with additional entries, groupadd cannot change information supplied by the network nameservice. However groupadd will verify the uniqueness of group name and group ID against the external nameservice.

NAME	groupdel – delete a group definition from the system
SYNOPSIS	<i>/usr/sbin/groupdel group</i>
AVAILABILITY	SUNWcsu
DESCRIPTION	The groupdel command deletes a group definition from the system. It deletes the appropriate entry from the /etc/group file.
OPTIONS	<i>group</i> A string of printable characters that specifies the group to be deleted.
FILES	/etc/group
SEE ALSO	users(1B), groupadd(1M), groupmod(1M), logins(1M), useradd(1M), userdel(1M), usermod(1M)
DIAGNOSTICS	The groupdel command exits with one of the following values: 0 Success. 2 Invalid command syntax. A usage message for the groupdel command is displayed. 6 <i>group</i> does not exist. 10 Cannot update the /etc/group file.
NOTES	groupdel only deletes a group definition that is in the local /etc/group file. If a network nameservice such as NIS or NIS+ is being used to supplement the local /etc/group file with additional entries, groupdel cannot change information supplied by the network nameservice.

NAME	groupmod – modify a group definition on the system
SYNOPSIS	<code>/usr/bin/groupmod [-g <i>gid</i> [-o]] [-n <i>name</i>] group</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	The groupmod command modifies the definition of the specified group by modifying the appropriate entry in the <code>/etc/group</code> file.
OPTIONS	<p>-g <i>gid</i> The group id for the new group. This group id must be a non-negative decimal integer below MAXUID as defined in <code>< param.h ></code>. The group ID defaults to the next available (unique) number above 99. (Group IDs from 0-99 are reserved by SunOS for future applications.)</p> <p>-o This option allows the <i>gid</i> to be duplicated (non-unique).</p> <p>-n <i>name</i> A string of printable characters that specifies a new name for the group. It may not include a colon (:) or newline (\n).</p> <p><i>group</i> The current name of the group to be modified.</p>
FILES	<code>/etc/group</code>
SEE ALSO	users(1B), groupadd(1M), groupdel(1M), logins(1M), useradd(1M), userdel(1M), usermod(1M)
DIAGNOSTICS	<p>The groupmod command exits with one of the following values:</p> <p>0 Success.</p> <p>2 Invalid command syntax. A usage message for the groupmod command is displayed.</p> <p>3 An invalid argument was provided to an option.</p> <p>4 <i>gid</i> is not unique (when the -o option is not used).</p> <p>6 <i>group</i> does not exist.</p> <p>9 <i>name</i> already exists as a group name.</p> <p>10 Cannot update the <code>/etc/group</code> file.</p>
NOTES	groupmod only modifies group definitions in the <code>/etc/group</code> file. If a network nameservice such as NIS or NIS+ is being used to supplement the local <code>/etc/group</code> file with additional entries, groupmod cannot change information supplied by the network nameservice. However groupmod will verify the uniqueness of group name and group id against the external nameservice.

NAME	halt, poweroff – stop the processor
SYNOPSIS	<code>/usr/sbin/halt [-lnqy]</code> <code>/usr/sbin/poweroff [-lnqy]</code>
AVAILABILITY	SUNWcsr
DESCRIPTION	halt and poweroff write out any pending information to the disks and then stop the processor. poweroff will have the machine remove power, if possible. halt and poweroff normally log the system shutdown to the system log daemon, syslogd (1M), and place a shutdown record in the login accounting file <code>/var/adm/wtmp</code> . These actions are inhibited if the <code>-n</code> or <code>-q</code> options are present.
OPTIONS	<code>-l</code> Suppress sending a message to the system log daemon, syslogd (1M), about who executed halt . <code>-n</code> Prevent the sync (1M) before stopping. <code>-q</code> Quick halt. No graceful shutdown is attempted. <code>-y</code> Halt the system, even from a dialup terminal.
FILES	<code>/var/adm/wtmp</code> login accounting file
SEE ALSO	init (1M), reboot (1M), shutdown (1M), sync (1M), syslogd (1M)
NOTES	halt does not execute the rc0 scripts as do shutdown (1M) and init (1M). poweroff is equivalent to init 5 .

NAME	hostconfig – configure a system’s host parameters
SYNOPSIS	<code>/usr/bin/hostconfig -p protocol [-d] [-n] [-v] [-i interface] [-f hostname]</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	The hostconfig program uses a network protocol to acquire a machine’s “host parameters” and then sets these parameters on the system. The program selects which protocol to use based on the argument to the required -p flag. Different protocols may set different host parameters. Currently, two protocols are defined but only one protocol is supported.
OPTIONS	<p>-p bootparams Use the “whoami” call of the RPC “bootparams” protocol. This sets the system’s <i>hostname</i>, <i>domainname</i>, and <i>default IP router</i> parameters.</p> <p>-p bootp Use the BOOTP protocol (not currently supported).</p> <p>-d Enable “debug” output.</p> <p>-n Run the network protocol, but do not set the acquired parameters into the system.</p> <p>-v Enable verbose output.</p> <p>-i interface Use only the named network interface to run the protocol.</p> <p>-f hostname Run the protocol as if this machine were named <i>hostname</i>.</p>
EXAMPLES	<p>To configure a machine’s host parameters using the “bootparams whoami” protocol with a verbose output, use</p> <pre>example% hostconfig -p bootparams -v</pre> <p>To see what parameters would be set using the “bootparams whoami” protocol, use</p> <pre>example% hostconfig -p bootparams -n -v</pre>
SEE ALSO	hostname(1) , domainname(1M) , route(1M)

NAME	htable – convert DoD Internet format host table
SYNOPSIS	<i>/usr/sbin/htable filename</i>
AVAILABILITY	SUNWnisu
DESCRIPTION	<p>htable converts a host table in the format specified by RFC 952 to the format used by the network library routines. Three files are created as a result of running htable: hosts, networks, and gateways. The hosts file is used by the gethostbyname(3N) routines in mapping host names to addresses. The networks file is used by the getnetbyname(3N) routines in mapping network names to numbers. The gateways file is used by the routing daemon to identify “passive” Internet gateways.</p> <p>If any of the files localhosts, localnetworks, or localgateways are present in the current directory, the file’s contents is prepended to the output file without interpretation. This allows sites to maintain local aliases and entries which are not normally present in the master database.</p> <p>htable is best used in conjunction with the gettable(1M) program which retrieves the DoD Internet host table from a host.</p>
FILES	localhosts localnetworks localgateways
SEE ALSO	gettable(1M) , gethostbyname(3N) , getnetbyname(3N) Harrenstien, Ken, Mary Stahl, and Elizabeth Feinler, <i>DoD Internet Host Table Specification</i> , RFC 952, Network Information Center, SRI International, Menlo Park, Calif., October 1985.
NOTES	Does not properly calculate the gateways file.

NAME	id – return user identity
SYNOPSIS	<pre> /usr/bin/id [user] /usr/bin/id -a [user] /usr/xpg4/bin/id [user] /usr/xpg4/bin/id -G [-n] [user] /usr/xpg4/bin/id -g [-nr] [user] /usr/xpg4/bin/id -u [-nr] [user] </pre>
AVAILABILITY	
/usr/bin/id	SUNWcsu, SUNWcar
/usr/xpg4/bin/id	SUNWxcu4
DESCRIPTION	<p>If no <i>user</i> operand is provided, the id utility will write the user and group IDs and the corresponding user and group names of the invoking process to standard output. If the effective and real IDs do not match, both will be written. If multiple groups are supported by the underlying system, the supplementary group affiliations of the invoking process also will be written.</p> <p>If a <i>user</i> operand is provided and the process has the appropriate privileges, the user and group IDs of the selected user will be written. In this case, effective IDs will be assumed to be identical to real IDs. If the selected user has more than one allowable group membership listed in the group database, these will be written in the same manner as the supplementary groups described in the preceding paragraph.</p>
Formats	<p>The following formats will be used when the LC_MESSAGES locale category specifies the "C" locale. In other locales, the strings uid, gid, eid, egid, and groups may be replaced with more appropriate strings corresponding to the locale.</p> <pre> "uid=%u(%s) gid=%u(%s)\n" <real user ID>, <user-name>, <real group ID>, <group-name> </pre> <p>If the effective and real user IDs do not match, the following will be inserted immediately before the \n character in the previous format:</p> <pre> " eid=%u(%s)" </pre> <p>with the following arguments added at the end of the argument list:</p> <pre> <effective user ID>, <effective user-name> </pre> <p>If the effective and real group IDs do not match, the following will be inserted directly before the \n character in the format string (and after any addition resulting from the effective and real user IDs not matching):</p> <pre> " egid=%u(%s)" </pre> <p>with the following arguments added at the end of the argument list:</p>

<effective group-ID>, <effective group name>

If the process has supplementary group affiliations or the selected user is allowed to belong to multiple groups, the first will be added directly before the NEWLINE character in the format string:

" groups=%u(%s)"

with the following arguments added at the end of the argument list:

<supplementary group ID>, <supplementary group name>

and the necessary number of the following added after that for any remaining supplementary group IDs:

",%u(%s)"

and the necessary number of the following arguments added at the end of the argument list:

<supplementary group ID>, <supplementary group name>

If any of the user ID, group ID, effective user ID, effective group ID or supplementary/multiple group IDs cannot be mapped by the system into printable user or group names, the corresponding (%s) and name argument will be omitted from the corresponding format string.

When any of the options are specified, the output format will be as described under **OPTIONS**.

OPTIONS

The following options are supported:

- a Reports user name, user ID and all the groups to which the user belongs.
- G Output all different group IDs (effective, real and supplementary) only, using the format "%u\n". If there is more than one distinct group affiliation, output each such affiliation, using the format " %u", before the newline character is output.
- g Output only the effective group ID, using the format "%u\n".
- n Output the name in the format "%s" instead of the numeric ID using the format "%u".
- r Output the real ID instead of the effective ID.
- u Output only the effective user ID, using the format "%u\n".

OPERANDS

The following operand is supported:

user The user (login) name for which information is to be written.

ENVIRONMENT

See **environ(5)** for descriptions of the following environment variables that affect the execution of **id**: **LC_CTYPE**, **LC_MESSAGES**, and **NLSPATH**.

EXIT STATUS

The following exit values are returned:

- 0** Successful completion.
- >0** An error occurred.

SEE ALSO `fold(1)`, `logname(1)`, `who(1)`, `getgid(2)`, `getgroups(2)`, `getuid(2)`, `environ(5)`

NOTES Output produced by the `-G` option and by the default case could potentially produce very long lines on systems that support large numbers of supplementary groups.

NAME	ifconfig – configure network interface parameters
SYNOPSIS	<pre> /sbin/ifconfig interface [address_family] [address [dest_address]] [up] [down] [auto-revarp] [netmask mask] [broadcast address] [metric n] [mtu n] [trailers -trailers] [private -private] [arp -arp] [plumb] [unplumb] /usr/sbin/ifconfig interface [address_family] [address [dest_address]] [up] [down] [auto-revarp] [netmask mask] [broadcast address] [metric n] [mtu n] [trailers -trailers] [private -private] [arp -arp] [plumb] [unplumb] </pre>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>ifconfig is used to assign an address to a network interface and/or to configure network interface parameters. ifconfig must be used at boot time to define the network address of each interface present on a machine; it may also be used at a later time to redefine an interface's address or other operating parameters. Used without options, ifconfig displays the current configuration for a network interface. If a protocol family is specified, ifconfig will report only the details specific to that protocol family. Only the super-user may modify the configuration of a network interface.</p> <p>The <i>interface</i> parameter is a string of the form <i>name physical-unit</i>, for example le0 or ie1, or of the form <i>name physical-unit : logical-unit</i>, for example le0:1. Three special interface names, -a, -ad and -au, are reserved and refer to all or a subset of the interfaces in the system. If one of these interface names is given, the commands following it are applied to all of the interfaces that match:</p> <ul style="list-style-type: none"> -a Apply the commands to all interfaces in the system. -ad Apply the commands to all “down” interfaces in the system. -au Apply the commands to all “up” interfaces in the system. <p>Since an interface may receive transmissions in differing protocols, each of which may require separate naming schemes, the parameters and addresses are interpreted according to the rules of some address family, specified by the <i>address_family</i> parameter. The address families currently supported are ether and inet. If no address family is specified, inet is assumed.</p> <p>For the TCP/IP family (inet), the address is either a host name present in the host name data base (see hosts(4)) or in the Network Information Service (NIS) map hosts, or a TCP/IP address expressed in the Internet standard “dot notation”. Typically, an Internet address specified in dot notation will consist of your system's network number and the machine's unique host number. A typical Internet address is 192.9.200.44, where 192.9.200 is the network number and 44 is the machine's host number.</p> <p>For the ether address family, the address is an Ethernet address represented as <i>x:x:x:x:x:x</i> where <i>x</i> is a hexadecimal number between 0 and FF.</p> <p>If the <i>dest_address</i> parameter is supplied in addition to the <i>address</i> parameter, it specifies the address of the correspondent on the other end of a point to point link.</p>

OPTIONS

arp	Enable the use of the Address Resolution Protocol in mapping between network level addresses and link level addresses (default). This is currently implemented for mapping between TCP/IP addresses and 10Mb/s Ethernet addresses.
-arp	Disable the use of the Address Resolution Protocol.
auto-revarp	Use the Reverse Address Resolution Protocol (RARP) to automatically acquire an address for this interface.
down	Mark an interface “down”. When an interface is marked “down”, the system will not attempt to transmit messages through that interface. If possible, the interface will be reset to disable reception as well. This action does not automatically disable routes using the interface.
plumb	Open the device associated with the physical interface name and setup the streams needed for TCP/IP to use the device. Before this is done the interface will not show up in the output of ifconfig -a .
unplumb	Destroy any streams associated with this device and close the device. After this command is executed the device name should not show up in the output of ifconfig -a .
private	Tells the in.routed routing daemon that the interface should not be advertised.
-private	Specify unadvertised interfaces.
trailers	This flag used to cause a non-standard encapsulation of inet packets on certain link levels. Drivers supplied with this release no longer use this flag, but it is ignored for compatibility.
-trailers	Disable the use of a “trailer” link level encapsulation.
up	Mark an interface “up”. This happens automatically when setting the first address on an interface. The up option enables an interface after an ifconfig down , reinitializing the hardware.
broadcast address	(inet only.) Specify the address to use to represent broadcasts to the network. The default broadcast address is the address with a host part of all 1’s. A + (plus sign) given for the broadcast value causes the broadcast address to be reset to a default appropriate for the (possibly new) address and netmask. Note: The arguments of ifconfig are interpreted left to right, and therefore <p style="text-align: center;">ifconfig -a netmask + broadcast +</p> <p style="text-align: center;">and</p> <p style="text-align: center;">ifconfig -a broadcast + netmask +</p> may result in different values being assigned for the interfaces’ broadcast addresses.

metric <i>n</i>	Set the routing metric of the interface to <i>n</i> , default 0 . The routing metric is used by the routing protocol. Higher metrics have the effect of making a route less favorable; metrics are counted as addition hops to the destination network or host.
mtu <i>n</i>	Set the maximum transmission unit of the interface to <i>n</i> . For many types of networks the mtu has an upper limit, for example, 1500 for Ethernet.
netmask <i>mask</i>	<p>(inet only.) Specify how much of the address to reserve for subdividing networks into sub-networks. The mask includes the network part of the local address and the subnet part, which is taken from the host field of the address. The mask contains 1's for the bit positions in the 32-bit address which are to be used for the network and subnet parts, and 0's for the host part. The mask should contain at least the standard network portion, and the subnet field should be contiguous with the network portion. The mask can be specified in one of four ways: a) with a single hexadecimal number with a leading 0x. b) with a dot-notation address, c) with a '+' (plus sign) address, or d) with a pseudo host name/pseudo network name listed in the network table networks(4). If a '+' (plus sign) is given for the netmask value, the mask is looked up in the netmasks database using the interface network number as the key.</p> <p>If a pseudo host name/pseudo network name is supplied as the netmask value, netmask data may be located in the hosts or networks table. ifconfig first looks up the name in the hosts table. If an entry is found, the host IP address is used as the netmask. If the entry isn't found there, ifconfig looks up the name in the networks table. (If the entry is found there, the IP network address is padded with a zero.) The hosts and netmasks tables are not designed for storage of netmasks; adding a netmask entry by using this option may confuse other programs, so we advise against using it. The system administrator may configure the source and lookup order in the tables via the name service switch. See nsswitch.conf(4) for more information.</p>

LOGICAL INTERFACES

Solaris TCP/IP allows multiple logical interfaces to be associated with a physical network interface. This allows a single machine to be assigned multiple IP addresses, even though it may have only one network interface. Physical network interfaces have names of the form *driver-name physical-unit-number*, while logical interfaces have names of the form *driver-name physical-unit-number logical-unit-number*. A physical interface is configured into the system using the **plumb** sub-command, for example:

```
ifconfig le0 plumb
```

Once a physical interface has been "plumbed", additional local interfaces can be configured by simply naming them in subsequent **ifconfig** commands. Logical interfaces do not need to be "plumbed". Simply mentioning their name in an **ifconfig** command is sufficient. For example, the command:

ifconfig le0:1

will allocate a logical interface associated with the physical interface le0. A logical interface can be configured with parameters (address, netmask, etc.) different from the physical interface that it is associated with. And logical interfaces that are associated with the same physical interface can be given different parameters as well. Each logical interface must be associated with a physical interface. So, for example, the logical interface le0:1 can only be configured after the physical interface le0 has been plumbed.

EXAMPLES

If your workstation is not attached to an Ethernet, the **le0** interface should be marked “down” as follows:

```
example% ifconfig le0 down
```

To print out the addressing information for each interface, use the following command:

```
example% ifconfig -a
```

To reset each interface’s broadcast address after the netmasks have been correctly set, use the next command:

```
example% ifconfig -a broadcast +
```

To change the Ethernet address for interface **le0**, use the following command:

```
example% ifconfig le0 ether aa:1:2:3:4:5
```

FILES

/etc/netmasks netmask data

SEE ALSO

in.routed(1M), **netstat(1M)**, **ethers(3N)**, **hosts(4)**, **netmasks(4)**, **networks(4)**, **nsswitch.conf(4)**, **arp(7P)**

DIAGNOSTICS

Messages indicating the specified interface does not exist, the requested address is unknown, or the user is not privileged and tried to alter an interface’s configuration.

NOTES

It is recommended that the names **broadcast**, **down**, **private**, **trailers**, **up**, and the other possible option names not be selected when choosing host names. Choosing any one of these names as host names will cause bizarre problems that can be extremely difficult to diagnose.

NAME	in.comsat, comsat – biff server
SYNOPSIS	/usr/sbin/in.comsat
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>comsat is the server process which listens for reports of incoming mail and notifies users who have requested to be told when mail arrives. It is invoked as needed by inetd(1M), and times out if inactive for a few minutes.</p> <p>comsat listens on a datagram port associated with the biff service specification (see services(4)) for one line messages of the form</p> <p style="text-align: center;"><i>user@mailbox-offset</i></p> <p>If the <i>user</i> specified is logged in to the system and the associated terminal has the owner execute bit turned on (by a biff y), the <i>offset</i> is used as a seek offset into the appropriate mailbox file, and the first 7 lines or 560 characters of the message are printed on the user's terminal. Lines which appear to be part of the message header other than the From, To, Date, or Subject lines are not printed when displaying the message.</p>
FILES	/var/adm/utmp to find out who is logged in and on what terminals
SEE ALSO	inetd (1M), services (4)
NOTES	The message header filtering is prone to error.

NAME	in.fingerd, fingerd – remote user information server										
SYNOPSIS	<code>/usr/sbin/in.fingerd</code>										
AVAILABILITY	SUNWcsu										
DESCRIPTION	<p>fingerd implements the server side of the Name/Finger protocol, specified in RFC 742. The Name/Finger protocol provides a remote interface to programs which display information on system status and individual users. The protocol imposes little structure on the format of the exchange between client and server. The client provides a single command line to the finger server which returns a printable reply.</p> <p>fingerd waits for connections on TCP port 79. Once connected, it reads a single command line terminated by RETURN-LINEFEED and passes the arguments to finger(1), prepended with <code>-s</code>. fingerd closes its connections as soon as the output is finished.</p> <p>You must invoke fingerd from inetd. See inetd(1M) for more information.</p>										
FILES	<table border="0"> <tr> <td><code>/var/adm/utmp</code></td> <td>user and accounting information</td> </tr> <tr> <td><code>/etc/passwd</code></td> <td>system password file</td> </tr> <tr> <td><code>/var/adm/lastlog</code></td> <td>last login times</td> </tr> <tr> <td><code>\$HOME/.plan</code></td> <td>user's plans</td> </tr> <tr> <td><code>\$HOME/.project</code></td> <td>user's projects</td> </tr> </table>	<code>/var/adm/utmp</code>	user and accounting information	<code>/etc/passwd</code>	system password file	<code>/var/adm/lastlog</code>	last login times	<code>\$HOME/.plan</code>	user's plans	<code>\$HOME/.project</code>	user's projects
<code>/var/adm/utmp</code>	user and accounting information										
<code>/etc/passwd</code>	system password file										
<code>/var/adm/lastlog</code>	last login times										
<code>\$HOME/.plan</code>	user's plans										
<code>\$HOME/.project</code>	user's projects										
SEE ALSO	<p>finger(1), inetd(1M)</p> <p>Harrenstien, Ken, <i>NAME/FINGER</i>, RFC 742, Network Information Center, SRI International, Menlo Park, Calif., December 1977.</p>										
NOTES	<p>Connecting directly to the server from a TIP or an equally narrow-minded TELNET-protocol user program can result in meaningless attempts at option negotiation being sent to the server, which will foul up the command line interpretation. fingerd should be taught to filter out IAC's and perhaps even respond negatively (IAC <i>will not</i>) to all option commands received.</p>										

NAME	in.ftpd, ftpd – file transfer protocol server
SYNOPSIS	in.ftpd [-dl] [-timeout]
AVAILABILITY	SUNWcsu
DESCRIPTION	in.ftpd is the Internet File Transfer Protocol (FTP) server process. The server is invoked by the Internet daemon inetd (1M) each time a connection to the FTP service (see services (4)) is made.
OPTIONS	<p>-d Debugging information is logged to the system log daemon syslogd(1M).</p> <p>-l Each FTP session is logged to the system log daemon syslogd(1M).</p> <p>-timeout Set the inactivity timeout period to <i>timeout</i>seconds. The FTP server will timeout an inactive session after 15 minutes.</p>
Requests	The FTP server currently supports the following FTP requests; case is not distinguished.
ABOR	abort previous command
ACCT	specify account (ignored)
ALLO	allocate storage (vacuously)
APPE	append to a file
CDUP	change to parent of current working directory
CWD	change working directory
DELE	delete a file
HELP	give help information
LIST	give list files in a directory (ls -lg)
MKD	make a directory
MODE	specify data transfer <i>mode</i>
NLST	give name list of files in directory (ls)
NOOP	do nothing
PASS	specify password
PASV	prepare for server-to-server transfer
PORT	specify data connection port
PWD	print the current working directory
QUIT	terminate session
RETR	retrieve a file
RMD	remove a directory
RNFR	specify rename-from file name
RNTO	specify rename-to file name

STOR	store a file
STOU	store a file with a unique name
STRU	specify data transfer <i>structure</i>
TYPE	specify data transfer <i>type</i>
USER	specify user name
XCUP	change to parent of current working directory
XCWD	change working directory
XMKD	make a directory
XPWD	print the current working directory
XRMD	remove a directory

The remaining FTP requests specified in RFC 959 are recognized, but not implemented.

The FTP server will abort an active file transfer only when the **ABOR** command is preceded by a Telnet “Interrupt Process” (IP) signal and a Telnet “Synch” signal in the command Telnet stream, as described in RFC 959.

in.ftpd interprets file names according to the “globbing” conventions used by **sh**(1). This allows users to utilize the metacharacters: * ? [] { } ~

in.ftpd authenticates users according to four rules.

- 1) The user name must be in the password data base, **/etc/passwd**, and have a password that is not null. A password must always be provided by the client before any file operations may be performed.
- 2) If the user name appears in the file **/etc/ftpusers**, **ftp** access is denied.
- 3) **ftp** access is denied if the user’s shell (from **/etc/passwd**) is not listed in the file **/etc/shells**. If the file **/etc/shells** does not exist, then the user’s shell must be one of the following:

```

/usr/bin/sh /usr/bin/csh /usr/bin/ksh
/usr/bin/jsh /bin/sh /bin/csh
/bin/ksh /bin/jsh /sbin/sh
/sbin/jsh

```

- 4) If the user name is “anonymous” or “ftp”, an entry for the user name *ftp* must be present in the password and shadow files. The user is then allowed to log in by specifying any password — by convention this is given as the user’s e-mail address (such as **user@host.Sun.COM**). Do not specify a valid shell in the password entry of the *ftp* user, and do not give it a valid password (use NP in the encrypted password field of the shadow file).

For anonymous ftp users, **in.ftpd** takes special measures to restrict the client’s access privileges. The server performs a **chroot**(2) command to the home directory of the “ftp” user. In order that system security is not breached, it is recommended that the “ftp” subtree be constructed with care; the following rules are suggested.

- ~ftp** Make the home directory owned by **root** and unwritable by anyone. This directory should not be on a file system mounted with the **nosuid** option.
- ~ftp/bin** Make this directory owned by the super-user and unwritable by anyone. Make this a symbolic link to **~ftp/usr/bin**. The program **ls(1)** must be present to support the list commands. This program should have mode 111.
- ~ftp/usr/lib** Make this directory owned by the super-user and unwritable by anyone. Copy the following shared libraries from **/usr/lib** into this directory.:
- ld.so***
 - libc.so***
 - libdl.so***
 - libintl.so***
 - libw.so***
 - libnsl.so***
 - libsocket.so***
 - nss_nis.so***
 - nss_nisplus.so***
 - nss_dns.so***
 - nss_files.so***
 - straddr.so***
- ~ftp/etc** Make this directory owned by the super-user and unwritable by anyone. Copies of the files **passwd(4)**, **group(4)**, and **netconfig(4)** must be present for the **ls(1)** command to work properly. These files should be mode 444.
- ~ftp/pub** Make this directory mode 777 and owned by **ftp**. Users should then place files which are to be accessible via the anonymous account in this directory.
- ~ftp/dev** Make this directory owned by the super-user and unwritable by anyone. First perform **ls -lL** on the device files listed below to determine their major and minor numbers, then use **mknod** to create them in this directory.
- /dev/zero**
 - /dev/tcp**
 - /dev/udp**
 - /dev/ticotsord**
- ~ftp/usr/share/lib/zoneinfo** Make this directory mode 555 and owned by the super-user. Copy its contents from **/usr/share/lib/zoneinfo**. This enables **ls -l** to display time and date stamps correctly.

EXAMPLES

To set up anonymous ftp, add the following entry to the `/etc/passwd` file. In this case, `/export/ftp` was chosen to be the anonymous ftp area, and the shell is the non-existent file `/nosuchshell`. This prevents users from logging in as the ftp user.

```
ftp:x:30000:30000:Anonymous FTP:/export/ftp:/nosuchshell
```

Add the following entry to `/etc/shadow`:

```
ftp:NP:6445::::::
```

The following is a shell script that will set up the anonymous ftp area. It presumes that names are resolved using NIS.

```
#!/bin/sh
# script to setup anonymous ftp area
#
# handle the optional command line argument
case $# in

  # the default location for the anon ftp comes from the passwd file
  0) ftphome="`grep '^ftp:' /etc/passwd | cut -d: -f6`"
    ::

  1) if [ "$1" = "start" ]; then
    ftphome="`grep '^ftp:' /etc/passwd | cut -d: -f6`"
    else
    ftphome=$1
    fi
    ::

  *) echo "Usage: $0 [anon-ftp-root]"
    exit 1
    ::
esac

if [ -z "${ftphome}" ]; then
  echo "$0: ftphome must be non-null"
  exit 2
fi

# This script assumes that ftphome is neither / nor /usr so ...
if [ "${ftphome}" = "/" -o "${ftphome}" = "/usr" ]; then
  echo "$0: ftphome must not be / or /usr"
  exit 2
fi

# If ftphome does not exist but parent does, create ftphome
if [ ! -d ${ftphome} ]; then
  # lack of -p below is intentional
```

```

mkdir ${ftphome}

fi
echo Setting up anonymous ftp area ${ftphome}

# Ensure that the /usr/bin directory exists
if [ ! -d ${ftphome}/usr/bin ]; then
    mkdir -p ${ftphome}/usr/bin
fi

cp /usr/bin/ls ${ftphome}/usr/bin
chmod 111 ${ftphome}/usr/bin/ls

# Now set the ownership and modes to match the man page
chown root ${ftphome}/usr/bin
chmod 555 ${ftphome}/usr/bin

# this may not be the right thing to do
# but we need the bin -> usr/bin link
if [ -r ${ftphome}/bin ]; then
    mv -f ${ftphome}/bin ${ftphome}/Obin
fi
ln -s usr/bin ${ftphome}

# Ensure that the /usr/lib and /etc directories exist
if [ ! -d ${ftphome}/usr/lib ]; then
    mkdir -p ${ftphome}/usr/lib
fi
if [ ! -d ${ftphome}/etc ]; then
    mkdir -p ${ftphome}/etc
fi

#Most of the following are needed for basic operation, except
#for libnsl.so, nss_nis.so, libsocket.so, and straddr.so which are
#needed to resolve NIS names.

cp /usr/lib/ld.so /usr/lib/ld.so.1 ${ftphome}/usr/lib

for lib in libc libdl libintl libw libnsl libsocket \
nss_nis nss_nisplus nss_dns nss_files
do
    cp /usr/lib/${lib}.so.1 ${ftphome}/usr/lib
    rm -f ${ftphome}/usr/lib/${lib}.so
    ln -s ./${lib}.so.1 ${ftphome}/usr/lib/${lib}.so
done

```

```

done

cp /usr/lib/straddr.so.2 ${ftphome}/usr/lib
rm -f ${ftphome}/usr/lib/straddr.so
ln -s ./straddr.so.2 ${ftphome}/usr/lib/straddr.so

cp /etc/passwd /etc/group /etc/netconfig ${ftphome}/etc

# Copy timezone database
mkdir -p ${ftphome}/usr/share/lib/zoneinfo
(cd ${ftphome}/usr/share/lib/zoneinfo
 (cd /usr/share/lib/zoneinfo; find . -print | cpio -o) | cpio -imdu
 find . -print | xargs chmod 555
 find . -print | xargs chown root
)

chmod 555 ${ftphome}/usr/lib/*
chmod 444 ${ftphome}/etc/*

# Now set the ownership and modes
chown root ${ftphome}/usr/lib ${ftphome}/etc
chmod 555 ${ftphome}/usr/lib ${ftphome}/etc

# Ensure that the /dev directory exists
if [ ! -d ${ftphome}/dev ]; then
  mkdir -p ${ftphome}/dev
fi

# make device nodes. ticotsord and udp are necessary for
# 'ls' to resolve NIS names.

for device in zero tcp udp ticotsord
do
  line='ls -lL /dev/${device} | sed -e 's/,/ /''
  major='echo $line | awk '{print $5}'
  minor='echo $line | awk '{print $6}'
  rm -f ${ftphome}/dev/${device}
  mknod ${ftphome}/dev/${device} c ${major} ${minor}
done

chmod 666 ${ftphome}/dev/*

## Now set the ownership and modes
chown root ${ftphome}/dev
chmod 555 ${ftphome}/dev

```

```
if [ ! -d ${ftphome}/pub ]; then
  mkdir -p ${ftphome}/pub
fi
chown ftp ${ftphome}/pub
chmod 777 ${ftphome}/pub
```

SEE ALSO

ftp(1), **ls(1)**, **aset(1M)**, **inetd(1M)**, **mknod(1M)**, **syslogd(1M)**, **chroot(2)**, **getsockopt(3N)**, **group(4)**, **inetd.conf(4)**, **netconfig(4)**, **netrc(4)**, **passwd(4)**, **services(4)**

Postel, Jon, and Joyce Reynolds, *File Transfer Protocol (FTP)*, RFC 959, Network Information Center, SRI International, Menlo Park, Calif., October 1985.

DIAGNOSTICS

in.ftpd logs various errors to **syslogd**, with a facility code of **daemon**.

Info Severity

These messages are logged only if the **-I** flag is specified.

FTPD: connection from *host* at *time*

A connection was made to **ftpd** from the host *host* at the date and time *time*.

FTPD: User *user* timed out after *timeout* seconds at *time*

The user *user* was logged out because they had not entered any commands after *timeout* seconds; the logout occurred at the date and time *time*.

Debug Severity

These messages are logged only if the **-d** flag is specified.

FTPD: command: *command*

A command line containing *command* was read from the FTP client.

lost connection

The FTP client dropped the connection.

<— *replycode*

<— *replycode*—

A reply was sent to the FTP client with the reply code *replycode*. The next message logged will include the message associated with the reply. If a — follows the reply code, the reply is continued on later lines.

NOTES

The anonymous account is inherently dangerous and should be avoided when possible.

The server must run as the super-user to create sockets with privileged port numbers. It maintains an effective user id of the logged in user, reverting to the super-user only when binding addresses to sockets. The possible security holes have been extensively scrutinized, but are possibly incomplete.

/etc/ftpusers contains a list of users who cannot access the system; the format of the file is one user name per line.

NAME	in.named, named, named-xfer – Internet domain name server
SYNOPSIS	in.named [-b <i>bootfile</i>] [-d <i>level</i>] [-p <i>port</i>] named-xfer
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>in.named is the Internet domain name server. It is used by hosts on the Internet to provide access to the Internet distributed naming database. See RFC 1034 and RFC 1035 for more details. With no arguments, in.named reads /etc/named.boot for any initial data, and listens for queries on a privileged port.</p> <p>named-xfer is called by in.named whenever in.named needs to perform a Zone Transfer. named-xfer should not be called independently.</p>
/etc/named.boot File Entries	<p>The following is a sample of /etc/named.boot file entries:</p> <pre> ; ; boot file for name server ; ; type domain source file or host ; domain berkeley.edu primary berkeley.edu named.db secondary cc.berkeley.edu 10.2.0.78 128.32.0.10 cache . named.ca </pre> <p>The domain line specifies that berkeley.edu is the domain of the given server.</p> <p>The primary line states that the file named.db contains authoritative data for berkeley.edu. The file named.db contains data in the Zone file format, described in RFC 1035, except that all domain names are relative to the origin; in this case, berkeley.edu (see Zone File Format below for a more detailed description).</p> <p>The secondary line specifies that all authoritative data under cc.berkeley.edu is to be transferred from the name server at 10.2.0.78. If the transfer fails it will try 128.32.0.10, and continue for up to ten tries at that address. The secondary copy is also authoritative for the domain.</p> <p>The cache line specifies that data in named.ca is to be placed in the cache (typically such data as the locations of root domain servers). The file named.ca is in the same format as named.db.</p>

Zone File Format

The Zone file consists of entries of the form:

```
$INCLUDE < filename >
$ORIGIN < domain >
< domain > < opt_ttl > < opt_class > < type > < resource_record_data >
```

where *domain* is "." for the root, "@" for the current origin, or a standard domain name. If *domain* is a standard domain name that does not end with ".", the current origin is appended to the domain. Domain names ending with "." are unmodified.

The *opt_ttl* field is an optional integer number for the time-to-live field. It defaults to zero.

The *opt_class* field is currently one token, **IN** for the Internet.

The *type* field is one of the following tokens; the data expected in the *resource_record_data* field is in parentheses.

A	A host address (dotted quad).
CNAME	The canonical name for an alias (domain).
HINFO	Host information (cpu_type OS_type).
MB	A mailbox domain name (domain).
MG	A mail group member (domain).
MINFO	Mailbox or mail list information (request_domain error_domain).
MR	A mail rename domain name (domain).
MX	A mail exchanger (domain).
NS	An authoritative name server (domain).
NULL	A null resource record (no format or data).
PTR	A domain name pointer (domain).
SOA	Marks the start of a zone of authority (5 numbers). See RFC 1035.
TXT	Arbitrary number of strings.
WKS	A well know service description (not implemented yet).

OPTIONS

-b <i>bootfile</i>	Use <i>bootfile</i> rather than /etc/named.boot .
-d <i>level</i>	Print debugging information. <i>level</i> is a number indicating the level of messages printed.
-p <i>port</i>	Use a different <i>port</i> number.

FILES

/etc/named.boot	name server configuration boot file
/etc/named.pid	the process ID
/var/tmp/named.run	debug output
/var/tmp/named_dump.db	dump of the name servers database

SEE ALSO**kill(1)**, **resolver(3N)**, **signal(3B)**, **resolv.conf(4)**

Braden, R. (Editor), *Requirements for Internet Hosts - Applications and Support*, RFC 1123, Internet Engineering Task Force - Network Working Group, October 1989

Mockapetris, Paul, *Domain Names - Concepts and Facilities*, RFC 1034, Network Information Center, SRI International, Menlo Park, Calif., November 1987.

Mockapetris, Paul, *Domain Names - Implementation and Specification*, RFC 1035, Network Information Center, SRI International, Menlo Park, Calif., November 1987.

Mockapetris, Paul, *Domain System Changes and Observations*, RFC 973, Network Information Center, SRI International, Menlo Park, Calif., January 1986.

Partridge, Craig, *Mail Routing and the Domain System*, RFC 974, Network Information Center, SRI International, Menlo Park, Calif., January 1986.

NOTES

The following signals have the specified effect when sent to the server process using the **kill(1)** command.

SIGHUP Reads **/etc/named.boot** and reloads database.

SIGINT Dumps the current database and cache to **/var/tmp/named_dump.db**.

SIGUSR1 Turns on debugging; each subsequent SIGUSR1 increments debug level.

SIGUSR2 Turns off debugging completely.

NAME	in.rarpd, rarpd – DARPA Reverse Address Resolution Protocol server
SYNOPSIS	<code>/usr/sbin/in.rarpd [-d] -a</code> <code>/usr/sbin/in.rarpd [-d] device unit</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>in.rarpd starts a daemon that responds to Reverse Address Resolution Protocol (RARP) requests. The daemon forks a copy of itself that runs in background. It must be run as root.</p> <p>RARP is used by machines at boot time to discover their Internet Protocol (IP) address. The booting machine provides its Ethernet address in a RARP request message. Using the ethers and hosts databases, in.rarpd maps this Ethernet address into the corresponding IP address which it returns to the booting machine in an RARP reply message. The booting machine must be listed in both databases for in.rarpd to locate its IP address. in.rarpd issues no reply when it fails to locate an IP address.</p> <p>in.rarpd uses the STREAMS-based Data Link Provider Interface (DLPI) message set to communicate directly with the datalink device driver.</p>
OPTIONS	<p>-a Get the list of available network interfaces from IP using the SIOCGIFADDR ioctl and start a RARP daemon process on each interface returned.</p> <p>-d Print assorted debugging messages while executing.</p>
EXAMPLES	<p>The following command starts an in.rarpd for each network interface name returned from <code>/dev/ip</code>:</p> <p style="padding-left: 40px;">example# /usr/sbin/in.rarpd -a</p> <p>The following command starts one in.rarpd on the device <code>/dev/le</code> with the device instance number 0.</p> <p style="padding-left: 40px;">example# /usr/sbin/in.rarpd le 0</p>
FILES	<p><code>/etc/ethers</code> file or NIS+ map</p> <p><code>/etc/hosts</code> file or NIS+ map</p> <p><code>/tftpboot</code></p> <p><code>/dev/ip</code></p> <p><code>/dev/arp</code></p>
SEE ALSO	<p>boot(1M), ifconfig(1M), ethers(4), hosts(4), netconfig(4), dlpi(7P)</p> <p>RFC-903, <i>A Reverse Address Resolution Protocol</i>, Network Information Center, SRI International.</p> <p>Unix International, <i>Data Link Provider Interface</i>, Version 2, May 7, 1991, Sun Microsystems, 800-6915-01.</p>

NAME	in.rdisc, rdisc – network router discovery daemon
SYNOPSIS	<pre>/usr/sbin/in.rdisc [-a] [-f] [-s] [send-address] [receive-address] /usr/sbin/in.rdisc -r [-p preference] [-T interval] [send-address] [receive-address]</pre>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>in.rdisc implements the ICMP router discovery protocol. The first form of the command is used on hosts and the second form is used on routers. On a host, in.rdisc is invoked at boot time to populate the network routing tables with default routes. On a router, it is also invoked at boot time in order to start advertising the router to all the hosts.</p>
Host (First Form)	<p>On a host, in.rdisc listens on the ALL_HOSTS (224.0.0.1) multicast address for ROUTER_ADVERTISE messages from routers. The received messages are handled by first ignoring those listed router addresses with which the host does not share a network. Among the remaining addresses, the ones with the highest preference are selected as default routers and a default route is entered in the kernel routing table for each one of them.</p> <p>Optionally, in.rdisc can avoid waiting for routers to announce themselves by sending out a few ROUTER_SOLICITATION messages to the ALL_ROUTERS (224.0.0.2) multicast address when it is started.</p> <p>A timer is associated with each router address. The address will no longer be considered for inclusion in the routing tables if the timer expires before a new <i>advertise</i> message is received from the router. The address will also be excluded from consideration if the host receives an <i>advertise</i> message with the preference being maximally negative.</p>
Router (Second Form)	<p>When in.rdisc is started on a router, it uses the SIOCGIFCONF ioctl(2) to find the interfaces configured into the system and it starts listening on the ALL_ROUTERS multicast address on all the interfaces that support multicast. It sends out <i>advertise</i> messages to the ALL_HOSTS multicast address advertising all its IP addresses. A few initial <i>advertise</i> messages are sent out during the first 30 seconds and after that it will transmit <i>advertise</i> messages approximately every 600 seconds.</p> <p>When in.rdisc receives a <i>solicitation</i> message, it sends an <i>advertise</i> message to the host that sent the <i>solicitation</i> message.</p> <p>When in.rdisc is terminated by a signal, it sends out an <i>advertise</i> message with the preference being maximally negative.</p>
OPTIONS	<p>-a Accept all routers independent of the preference they have in their <i>advertise</i> messages. Normally, in.rdisc only accepts (and enters in the kernel routing tables) the router or routers with the highest preference.</p>

- f** Run **in.rdisc** forever even if no routers are found. Normally, **in.rdisc** gives up if it has not received any *advertise* message after soliciting three times, in which case it exits with a non-zero exit code. If **-f** is not specified in the first form then **-s** must be specified.
- r** Act as a router, rather than a host.
- s** Send three *solicitation* messages initially to quickly discover the routers when the system is booted. When **-s** is specified, **in.rdisc** exits with a non-zero exit code if it can not find any routers. This can be overridden with the **-f** option.
- p preference** Set the preference transmitted in the *solicitation* messages. The default is zero.
- T interval** Set the interval between transmitting the *advertise* messages. The default time is 600 seconds.

SEE ALSO**in.routed(1M), ioctl(2), icmp(7P), inet(7P)**

Deering, S.E., ed "ICMP Router Discovery Messages," RFC 1256, Network Information Center, SRI International, Menlo Park, Calif., September 1991.

NAME	in.rexecd, rexecd – remote execution server
SYNOPSIS	in.rexecd
DESCRIPTION	<p>in.rexecd is the server for the rexec(3N) routine. The server provides remote execution facilities with authentication based on user names and passwords. It is invoked automatically as needed by inetd(1M), and then executes the following protocol:</p> <ol style="list-style-type: none"> 1) The server reads characters from the socket up to a null (\0) byte. The resultant string is interpreted as an ASCII number, base 10. 2) If the number received in step 1 is non-zero, it is interpreted as the port number of a secondary stream to be used for the stderr. A second connection is then created to the specified port on the client's machine. 3) A null terminated user name of at most 16 characters is retrieved on the initial socket. 4) A null terminated password of at most 16 characters is retrieved on the initial socket. 5) A null terminated command to be passed to a shell is retrieved on the initial socket. The length of the command is limited by the upper bound on the size of the system's argument list. 6) rexecd then validates the user as is done at login time and, if the authentication was successful, changes to the user's home directory, and establishes the user and group protections of the user. If any of these steps fail the connection is aborted and a diagnostic message is returned. 7) A null byte is returned on the connection associated with the stderr and the command line is passed to the normal login shell of the user. The shell inherits the network connections established by rexecd.
SEE ALSO	inetd(1M) , rexec(3N)
DIAGNOSTICS	<p>All diagnostic messages are returned on the connection associated with the stderr, after which any network connections are closed. An error is indicated by a leading byte with a value of 1 (0 is returned in step 7 above upon successful completion of all the steps prior to the command execution).</p> <p>username too long The name is longer than 16 characters.</p> <p>password too long The password is longer than 16 characters.</p> <p>command too long The command line passed exceeds the size of the argument list (as configured into the system).</p>

Login incorrect.

No password file entry for the user name existed.

Password incorrect.

The wrong password was supplied.

No remote directory.

The **chdir** command to the home directory failed.

Try again.

A **fork** by the server failed.

/usr/bin/sh: ...

The user's login shell could not be started.

NAME	in.rlogind, rlogind – remote login server
SYNOPSIS	/usr/sbin/in.rlogind
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>in.rlogind is the server for the rlogin(1) program. The server provides a remote login facility with authentication based on privileged port numbers.</p> <p>in.rlogind is invoked by inetd(1M) when a remote login connection is established, and executes the following protocol:</p> <ul style="list-style-type: none"> • The server checks the client's source port. If the port is not in the range 0-1023, the server aborts the connection. • The server checks the client's source address. If an entry for the client exists in both /etc/hosts and /etc/hosts.equiv, a user logging in from the client is not prompted for a password. If the address is associated with a host for which no corresponding entry exists in /etc/hosts, the user is prompted for a password, regardless of whether or not an entry for the client is present in /etc/hosts.equiv (see hosts(4) and hosts.equiv(4)). <p>Once the source port and address have been checked, in.rlogind allocates a pseudo-terminal and manipulates file descriptors so that the slave half of the pseudo-terminal becomes the stdin, stdout, and stderr for a login process. The login process is an instance of the login(1) program, invoked with the -r option. The login process then proceeds with the authentication process as described in in.rshd(1M); but if automatic authentication fails, it reprompts the user to login.</p> <p>The parent of the login process manipulates the master side of the pseudo-terminal, operating as an intermediary between the login process and the client instance of the rlogin program. In normal operation, a packet protocol is invoked to provide Ctrl-S/ Ctrl-Q type facilities and propagate interrupt signals to the remote programs. The login process propagates the client terminal's baud rate and terminal type, as found in the environment variable, TERM; see environ(4).</p>
SEE ALSO	login(1), rlogin(1), in.rshd(1M), inetd(1M), environ(4), hosts(4), hosts.equiv(4)
DIAGNOSTICS	<p>All diagnostic messages are returned on the connection associated with the stderr, after which any network connections are closed. An error is indicated by a leading byte with a value of 1.</p> <p>Hostname for your address unknown. No entry in the host name database existed for the client's machine.</p> <p>Try again. A <i>fork</i> by the server failed.</p>

/usr/bin/sh: . . .

The user's login shell could not be started.

NOTES

The authentication procedure used here assumes the integrity of each client machine and the connecting medium. This is insecure, but is useful in an "open" environment.

A facility to allow all data exchanges to be encrypted should be present.

NAME	in.routed, routed – network routing daemon
SYNOPSIS	<code>/usr/sbin/in.routed [-s] [-q] [-t] [-g] [-S] [logfile]</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>in.routed is invoked at boot time to manage the network routing tables. The routing daemon uses a variant of the Xerox NS Routing Information Protocol in maintaining up-to-date kernel routing table entries.</p> <p>In normal operation, in.routed listens on udp(7P) socket 520 (decimal) for routing information packets. If the host is an internetwork router, it periodically supplies copies of its routing tables to any directly connected hosts and networks.</p> <p>When in.routed is started, it uses the SIOCGIFCONF ioctl(2) to find those directly connected interfaces configured into the system and marked “up” (the software loopback interface is ignored). If multiple interfaces are present, it is assumed the host will forward packets between networks. in.routed then transmits a <i>request</i> packet on each interface (using a broadcast packet if the interface supports it) and enters a loop, listening for <i>request</i> and <i>response</i> packets from other hosts.</p> <p>When a <i>request</i> packet is received, in.routed formulates a reply based on the information maintained in its internal tables. The <i>response</i> packet contains a list of known routes, each marked with a “hop count” metric (a count of 16, or greater, is considered “infinite”). The metric associated with each route returned, provides a metric relative to the sender.</p> <p><i>request</i> packets received by in.routed are used to update the routing tables if one of the following conditions is satisfied:</p> <ul style="list-style-type: none"> • No routing table entry exists for the destination network or host, and the metric indicates the destination is “reachable” (that is, the hop count is not infinite). • The source host of the packet is the same as the router in the existing routing table entry. That is, updated information is being received from the very internetwork router through which packets for the destination are being routed. • The existing entry in the routing table has not been updated for some time (defined to be 90 seconds) and the route is at least as cost effective as the current route. • The new route describes a shorter route to the destination than the one currently stored in the routing tables; the metric of the new route is compared against the one stored in the table to decide this. <p>When an update is applied, in.routed records the change in its internal tables and generates a <i>response</i> packet to all directly connected hosts and networks. in.routed waits a short period of time (no more than 30 seconds) before modifying the kernel’s routing tables to allow possible unstable situations to settle.</p> <p>In addition to processing incoming packets, in.routed also periodically checks the routing table entries. If an entry has not been updated for 3 minutes, the entry’s metric is set to infinity and marked for deletion. Deletions are delayed an additional 60 seconds to</p>

insure the invalidation is propagated throughout the internet.

Hosts acting as internetwork routers gratuitously supply their routing tables every 30 seconds to all directly connected hosts and networks.

In addition to the facilities described above, **in.routed** supports the notion of “distant” **passive** and **active** gateways. When **in.routed** is started up, it reads the file **gateways** to find gateways which may not be identified using the **SIOCGIFCONF ioctl**. Gateways specified in this manner should be marked **passive** if they are not expected to exchange routing information, while gateways marked **active** should be willing to exchange routing information (that is, they should have a **in.routed** process running on the machine). Passive gateways are maintained in the routing tables forever. Information regarding their existence is not included in any routing information transmitted. Active gateways are treated equally to network interfaces. Routing information is distributed to the gateway and if no routing information is received for a period of time, the associated route is deleted.

The **gateways** is comprised of a series of lines, each in the following format:

```
< net | host > filename1 gateway filename2 metric value < passive | active >
```

The **net** or **host** keyword indicates if the route is to a network or specific host.

filename1 is the name of the destination network or host. This may be a symbolic name located in **networks** or **hosts**, or an Internet address specified in “dot” notation; see **inet(3N)**.

filename2 is the name or address of the gateway to which messages should be forwarded.

value is a metric indicating the hop count to the destination host or network.

The keyword **passive** or **active** indicates if the gateway should be treated as passive or active (as described above).

OPTIONS

- g** Is used on internetwork routers to offer a route to the “default” destination. This is typically used on a gateway to the Internet, or on a gateway that uses another routing protocol whose routes are not reported to other local routers.
- q** Is the opposite of the **-s** option.
- s** Forces **in.routed** to supply routing information whether it is acting as an internetwork router or not.
- S** If **in.routed** is not acting as an internetwork router it will, instead of entering the whole routing table in the kernel, only enter a default route for each internetwork router. This reduces the the memory requirements without losing any routing reliability.

-t All packets sent or received are printed on standard output. In addition, **in.routed** will not divorce itself from the controlling terminal so that interrupts from the keyboard will kill the process. Any other argument supplied is interpreted as the name of the file in which **in.routed**'s actions should be logged. This log contains information about any changes to the routing tables and a history of recent messages sent and received which are related to the changed route.

FILES

/etc/gateways	for distant gateways
/etc/networks	associations of Internet Protocol network numbers with network names
/etc/hosts	Internet host table

SEE ALSO **ioctl(2)**, **inet(7P)**, **udp(7P)**

NOTES

The kernel's routing tables may not correspond to those of **in.routed** for short periods of time while processes that utilize existing routes exit; the only remedy for this is to place the routing process in the kernel.

in.routed should listen to intelligent interfaces, such as an IMP, and to error protocols, such as ICMP, to gather more information.

NAME	in.rshd, rshd – remote shell server
SYNOPSIS	in.rshd <i>host.port</i>
DESCRIPTION	<p>in.rshd is the server for the rsh(1) program. The server provides remote execution facilities with authentication based on privileged port numbers.</p> <p>in.rshd is invoked by inetd(1M) each time a shell service is requested, and executes the following protocol:</p> <ol style="list-style-type: none"> 1) The server checks the client's source port. If the port is not in the range 0-1023, the server aborts the connection. The client's host address (in hex) and port number (in decimal) are the arguments passed to in.rshd. 2) The server reads characters from the socket up to a null (\0) byte. The resultant string is interpreted as an ASCII number, base 10. 3) If the number received in step 1 is non-zero, it is interpreted as the port number of a secondary stream to be used for the stderr. A second connection is then created to the specified port on the client's machine. The source port of this second connection is also in the range 0-1023. 4) The server checks the client's source address. If the address is associated with a host for which no corresponding entry exists in the host name data base (see hosts(4)), the server aborts the connection. 5) A null terminated user name of at most 16 characters is retrieved on the initial socket. This user name is interpreted as a user identity to use on the <i>server's</i> machine. 6) A null terminated user name of at most 16 characters is retrieved on the initial socket. This user name is interpreted as the user identity on the <i>client's</i> machine. 7) A null terminated command to be passed to a shell is retrieved on the initial socket. The length of the command is limited by the upper bound on the size of the system's argument list. 8) in.rshd then validates the user according to the following steps. The remote user name is looked up in the password file and a chdir is performed to the user's home directory. If the lookup fails, the connection is terminated. If the chdir fails, it does a chdir to / (root). If the user is not the super-user, (user ID 0), the file /etc/hosts.equiv is consulted for a list of hosts considered "equivalent". If the client's host name is present in this file, the authentication is considered successful. If the lookup fails, or the user is the super-user, then the file .rhosts in the home directory of the remote user is checked for the machine name and identity of the user on the client's machine. If this lookup fails, the connection is terminated. 9) A null byte is returned on the connection associated with the stderr and the command line is passed to the normal login shell of the user. (The PATH variable is set to /usr/bin.) The shell inherits the network connections established by in.rshd.

FILES	/etc/hosts.equiv
SEE ALSO	rsh(1), inetd(1M), hosts(4)
DIAGNOSTICS	<p>The following diagnostic messages are returned on the connection associated with stderr, after which any network connections are closed. An error is indicated by a leading byte with a value of 1 in step 9 above (0 is returned above upon successful completion of all the steps prior to the command execution).</p> <p>locuser too long The name of the user on the client's machine is longer than 16 characters.</p> <p>remuser too long The name of the user on the remote machine is longer than 16 characters.</p> <p>command too long The command line passed exceeds the size of the argument list (as configured into the system).</p> <p>Hostname for your address unknown. No entry in the host name database existed for the client's machine.</p> <p>Login incorrect. No password file entry for the user name existed.</p> <p>Permission denied. The authentication procedure described above failed.</p> <p>Can't make pipe. The pipe needed for the stderr was not created.</p> <p>Try again. A <i>fork</i> by the server failed.</p>
NOTES	<p>The authentication procedure used here assumes the integrity of each client machine and the connecting medium. This is insecure, but is useful in an "open" environment. A facility to allow all data exchanges to be encrypted should be present.</p>

NAME	in.rwhod, rwhod – system status server
SYNOPSIS	<code>/usr/sbin/in.rwhod [-m [ttl]]</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>in.rwhod is the server which maintains the database used by the rwho(1) and ruptime(1) programs. Its operation is predicated on the ability to broadcast or multicast messages on a network.</p> <p>in.rwhod operates as both a producer and consumer of status information. As a producer of information it periodically queries the state of the system and constructs status messages which are broadcast or multicast on a network. As a consumer of information, it listens for other in.rwhod servers' status messages, validating them, then recording them in a collection of files located in the directory <code>/var/spool/rwho</code>.</p> <p>The rwho server transmits and receives messages at the port indicated in the rwho service specification, see services(4). The messages sent and received, are of the form:</p> <pre> struct outmp { char out_line[8]; /* tty name */ char out_name[8]; /* user id */ long out_time; /* time on */ }; struct whod { char wd_vers; char wd_type; char wd_fill[2]; int wd_sendtime; int wd_recvtime; char wd_hostname[32]; int wd_loadav[3]; int wd_boottime; struct whoent { struct outmp we_utmp; int we_idle; } wd_we[1024 / sizeof (struct whoent)]; }; </pre> <p>All fields are converted to network byte order prior to transmission. The load averages are as calculated by the w(1) program, and represent load averages over the 5, 10, and 15 minute intervals prior to a server's transmission. The host name included is that returned by the uname(2) system call. The array at the end of the message contains information about the users who are logged in to the sending machine. This information includes the contents of the utmp(4) entry for each non-idle terminal line and a value indicating the time since a character was last received on the terminal line.</p>

Messages received by the **rwho** server are discarded unless they originated at a **rwho** server's port. In addition, if the host's name, as specified in the message, contains any unprintable ASCII characters, the message is discarded. Valid messages received by **in.rwhod** are placed in files named **whod.hostname** in the directory **/var/spool/rwho**. These files contain only the most recent message, in the format described above.

Status messages are generated approximately once every 60 seconds. **in.rwhod** performs an **nlist(3E)** on **/dev/ksyms** every 10 minutes to guard against the possibility that this file is not the system image currently operating.

OPTIONS **-m [ttl]** Use the rwho IP multicast address (224.0.1.3) when transmitting. Receive announcements both on this multicast address and on the IP broadcast address. If *ttl* is not specified **in.rwhod** will multicast on all interfaces but with the IP TimeToLive set to 1 (that is, packets will not be forwarded by multicast routers.) If *ttl* is specified **in.rwhod** will only transmit packets on one interface and setting the IP TimeToLive to the specified *ttl*.

FILES **/dev/ksyms** system namelist
/var/spool/rwho/rwhod/*
 information about other machines

SEE ALSO **runtime(1), rwho(1), w(1), uname(2), nlist(3E), services(4), utmp(4)**

WARNINGS This service can cause network performance problems when used by several hosts on the network. It is not run at most sites by default. If used, include the multicast option (**-m**).

NOTES This service takes up progressively more network bandwidth as the number of hosts on the local net increases. For large networks, the cost becomes prohibitive.
in.rwhod should relay status information between networks. People often interpret the server dying as a machine going down.

NAME	in.talkd, talkd – server for talk program
SYNOPSIS	in.talkd
AVAILABILITY	SUNWcsu
DESCRIPTION	talkd is a server used by the talk(1) program. It listens at the UDP port indicated in the “talk” service description; see services(4) . The actual conversation takes place on a TCP connection that is established by negotiation between the two machines involved.
SEE ALSO	talk(1) , inetd(1M) , services(4)
NOTES	The protocol is architecture dependent.

NAME	in.telnetd, telnetd – DARPA TELNET protocol server
SYNOPSIS	<code>/usr/sbin/in.telnetd</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>in.telnetd is a server that supports the DARPA standard TELNET virtual terminal protocol. in.telnetd is invoked in the internet server (see inetd(1M)), normally for requests to connect to the TELNET port as indicated by the <code>/etc/services</code> file (see services(4)).</p> <p>in.telnetd operates by allocating a pseudo-terminal device for a client, then creating a login process which has the slave side of the pseudo-terminal as its standard input, output, and error. in.telnetd manipulates the master side of the pseudo-terminal, implementing the TELNET protocol and passing characters between the remote client and the login process.</p> <p>When a TELNET session starts up, in.telnetd sends TELNET options to the client side indicating a willingness to do <i>remote echo</i> of characters, and to <i>suppress go ahead</i>. The pseudo-terminal allocated to the client is configured to operate in “cooked” mode, and with XTABS, ICRNL, and ONLCR enabled (see termio(7I)).</p> <p>in.telnetd is willing to do: <i>echo</i>, <i>binary</i>, <i>suppress go ahead</i>, and <i>timing mark</i>. in.telnetd is willing to have the remote client do: <i>binary</i>, <i>terminal type</i>, <i>terminal size</i>, <i>logout option</i>, and <i>suppress go ahead</i>.</p> <p>in.telnetd also allows environment variables to be passed, provided that the client negotiates this during the initial option negotiation. The DISPLAY environment variable may be sent this way, either by the TELNET general environment passing methods, or via the XDISPLOC TELNET option. DISPLAY can be passed in the environment option during the same negotiation where XDISPLOC is used. Note that if you use both methods, use the same value for both. Otherwise, the results may be unpredictable.</p> <p>These options are specified in Internet standards RFC 1096, RFC 1408, RFC 1571, and RFC 1572.</p>
SEE ALSO	<p>telnet(1), services(4), termio(7I)</p> <p>Alexander, S., “<i>TELNET Environment Option</i>,” RFC 1572, Network Information Center, SRI International, Menlo Park, Calif., January 1994.</p> <p>Borman, Dave, “<i>TELNET Environment Option</i>,” RFC 1408, Network Information Center, SRI International, Menlo Park, Calif., January 1993.</p> <p>Borman, Dave, “<i>TELNET Environment Option Interoperability Issues</i>,” RFC 1571, Network Information Center, SRI International, Menlo Park, Calif., January 1994.</p> <p>Crispin, Mark, “<i>TELNET Logout Option</i>,” RFC 727, Network Information Center, SRI International, Menlo Park, Calif., April 1977.</p> <p>Marcy, G., “<i>TELNET X Display Location Option</i>,” RFC 1096, Network Information Center, SRI International, Menlo Park, Calif., March 1989.</p>

Postel, Jon, and Joyce Reynolds, “*TELNET Protocol Specification*,” RFC 854, Network Information Center, SRI International, Menlo Park, Calif., May 1983.

Waitzman, D., “*TELNET Window Size Option*,” RFC 1073, Network Information Center, SRI International, Menlo Park, Calif., October 1988.

NOTES

Some TELNET commands are only partially implemented.

Binary mode has no common interpretation except between similar operating systems

The terminal type name received from the remote client is converted to lower case.

The *packet* interface to the pseudo-terminal should be used for more intelligent flushing of input and output queues.

in.telnetd never sends TELNET *go ahead* commands.

NAME	in.tftpd, tftpd – Internet Trivial File Transfer Protocol server
SYNOPSIS	in.tftpd [-s] [<i>homedir</i>]
DESCRIPTION	<p>tftpd is a server that supports the Internet Trivial File Transfer Protocol (TFTP). This server is normally started by inetd(1M) and operates at the port indicated in the tftp Internet service description in the /etc/inetd.conf file. By default, the entry for in.tftpd in etc/inetd.conf is commented out. To make in.tftpd operational, the comment character(s) must be deleted from the file. See inetd.conf(4).</p> <p>Before responding to a request, the server attempts to change its current directory to <i>homedir</i>; the default directory is /tftpboot.</p> <p>The use of tftp does not require an account or password on the remote system. Due to the lack of authentication information, in.tftpd will allow only publicly readable files to be accessed. Files may be written only if they already exist and are publicly writable. Note that this extends the concept of “public” to include all users on all hosts that can be reached through the network; this may not be appropriate on all systems, and its implications should be considered before enabling this service.</p> <p>in.tftpd runs with the user ID and group ID set to [G]UID_NOBODY under the assumption that no files exist with that owner or group. However, nothing checks this assumption or enforces this restriction.</p>
OPTIONS	-s Secure. When specified, the directory change to <i>homedir</i> must succeed. The daemon also changes its root directory to <i>homedir</i> .
FILES	/etc/inetd.conf
SEE ALSO	tftp (1), inetd (1M), netconfig (4) Sollins, K.R., <i>The TFTP Protocol (Revision 2)</i> , RFC 783, Network Information Center, SRI International, Menlo Park, Calif., June 1981.

NAME	in.tnamed, tnamed – DARPA trivial name server
SYNOPSIS	<code>/usr/sbin/in.tnamed [-v]</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	in.tnamed is a server that supports the DARPA Name Server Protocol. The name server operates at the port indicated in the “name” service description (see services(4)), and is invoked by inetd(1M) when a request is made to the name server.
OPTIONS	-v Invoke the daemon in verbose mode.
SEE ALSO	uucp(1C) , inetd(1M) , services(4) Postel, Jon, <i>Internet Name Server</i> , IEN 116, SRI International, Menlo Park, California, August 1979.
NOTES	The protocol implemented by this program is obsolete. Its use should be phased out in favor of the Internet Domain Name Service (DNS) protocol.

NAME	in.uucpd, uucpd – UUCP server						
SYNOPSIS	<code>/usr/sbin/in.uucpd [-n]</code>						
AVAILABILITY	SUNWbnuu						
DESCRIPTION	<p>in.uucpd is the server for supporting UUCP connections over networks.</p> <p>in.uucpd is invoked by inetd(1M) when a UUCP connection is established (that is, a connection to the port indicated in the “uucp” service specification; see services(4)), and executes the following protocol:</p> <ol style="list-style-type: none"> 1) The server prompts with login:. The uucico(1M) process at the other end must supply a username. 2) Unless the username refers to an account without a password, the server then prompts with Password:. The uucico process at the other end must supply the password for that account. <p>If the username is not valid, or is valid but refers to an account that does not have <code>/usr/lib/uucp/uucico</code> as its login shell, or if the password is not the correct password for that account, the connection is dropped. Otherwise, uucico is run, with the user ID, group ID, group set, and home directory for that account, with the environment variables USER and LOGNAME set to the specified username, and with a -u flag specifying the username. Unless the -n flag is specified, entries are made in <code>/var/adm/utmp</code>, <code>/var/adm/wtmp</code>, and <code>/var/adm/lastlog</code> for the username. in.uucpd must be invoked by a user with appropriate privilege (usually root) in order to be able to verify that the password is correct.</p>						
FILES	<table border="0"> <tr> <td><code>/var/adm/utmp</code></td> <td>accounting</td> </tr> <tr> <td><code>/var/adm/wtmp</code></td> <td>accounting</td> </tr> <tr> <td><code>/var/adm/lastlog</code></td> <td>time of last login</td> </tr> </table>	<code>/var/adm/utmp</code>	accounting	<code>/var/adm/wtmp</code>	accounting	<code>/var/adm/lastlog</code>	time of last login
<code>/var/adm/utmp</code>	accounting						
<code>/var/adm/wtmp</code>	accounting						
<code>/var/adm/lastlog</code>	time of last login						
SEE ALSO	inetd (1M), uucico (1M), services (4)						
DIAGNOSTICS	<p>All diagnostic messages are returned on the connection, after which the connection is closed.</p> <p>user read An error occurred while reading the username.</p> <p>passwd read An error occurred while reading the password.</p> <p>Login incorrect. The username is invalid or refers to an account with a login shell other than <code>/usr/lib/uucp/uucico</code>, or the password is not the correct password for the account.</p>						

NAME	inetd – Internet services daemon
SYNOPSIS	inetd [-d] [-s] [-t] [-r count interval] [<i>configuration-file</i>]
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>inetd is the server process for the Internet standard services. It is usually started up at system boot time. The <i>configuration-file</i> lists the services that inetd is to provide. If no <i>configuration-file</i> is given on the command line, inetd reads its configuration information from the file <code>/etc/inetd.conf</code>. See inetd.conf(4) for more information on the format of this file. inetd listens for service requests on the TCP or UDP ports associated with each of the service listed in the configuration file. When a request arrives, inetd executes the server program associated with the service.</p> <p>A service can be configured to be “single-threaded”, in which case inetd waits for the server process to exit before starting a second server process. RPC services can also be started by inetd.</p> <p>inetd provides a number of simple Internet services internally. These include echo, discard, chargen (character generator), daytime (human-readable time), and time (machine-readable time, in the form of the number of seconds since midnight, January 1, 1900).</p> <p>inetd rereads its configuration file once when it is started and again whenever it receives a hangup signal, SIGHUP. New services can be activated, and existing services deleted or modified by editing the configuration file, then sending inetd a SIGHUP signal.</p>
OPTIONS	<p>-d Runs inetd in the foreground and enables debugging output.</p> <p>-s Allows you to run inetd “stand-alone,” outside the Service Access Facility (SAF). If the -s option is omitted, inetd will attempt to contact the service access controller (SAC) and will exit if SAC is not already running. See sac(1M).</p> <p>-t Instructs inetd to trace the incoming connections for all of its TCP services. It does this by logging the client’s IP address and TCP port number, along with the name of the service, using the syslog(3) facility. UDP services can not be traced. When tracing is enabled, inetd uses the syslog facility code “daemon” and “notice” priority level.</p> <p>-r Allows inetd to detect and then suspend “broken” servers. The -r flag has the form:</p> <p style="padding-left: 2em;">-r count interval</p> <p><i>count</i> and <i>interval</i> are decimal numbers that represent the maximum <i>count</i> of invocations per <i>interval</i> of seconds a service may be started before the service is considered “broken”.</p> <p>Once considered “broken”, a server is suspended for ten minutes. After ten minutes, inetd again enables service, hoping the server behaves correctly.</p>

If the **-r** flag is not specified, **inetd** behaves as though **-r40 60** was specified.

OPERANDS

configuration-file

Lists the services **inetd** is to provide.

EXIT STATUS

inetd does not return an Exit Status.

SEE ALSO

in.ftpd(1M), **in.rexecd(1M)**, **in.rshd(1M)**, **in.tftpd(1M)**, **sac(1M)**, **inetd.conf(4)**

Postel, Jon, "Echo Protocol," RFC 862, Network Information Center, SRI International, Menlo Park, CA, May 1983.

Postel, Jon, "Discard Protocol," RFC 863, Network Information Center, SRI International, Menlo Park, CA, May 1983.

Postel, Jon, "Character Generator Protocol," RFC 864, Network Information Center, SRI International, Menlo Park, CA, May 1983.

Postel, Jon, "Daytime Protocol," RFC 867, Network Information Center, SRI International, Menlo Park, CA, May 1983.

Postel, Jon, and Ken Harrenstien, "Time Protocol," RFC 868, Network Information Center, SRI International, Menlo Park, CA, May 1983.

WARNINGS

Do not configure **udp** services as **nowait**. This will cause a race condition where the **inetd** program selects on the socket and the server program reads from the socket. Many server programs will be forked and performance will be severely compromised.

NOTES

For RPC services, **inetd** listens on all the transports (not only **tcp** and **udp**) as specified for each service in the **inetd.conf(4)** file.

NAME	infocmp – compare or print out terminfo descriptions
SYNOPSIS	<code>/usr/bin/infocmp [-d] [-c] [-n] [-I] [-L] [-C] [-r] [-u] [-s d i l c] [-v] [-V] [-1] [-w width] [-A directory] [-B directory] [termname ...]</code>
DESCRIPTION	infocmp compares a binary terminfo entry with other terminfo entries, rewrites a terminfo description to take advantage of the use= terminfo field, or prints out a terminfo description from the binary file (term) in a variety of formats. It displays boolean fields first, then numeric fields, followed by the string fields. If no options are specified and zero, or one <i>termname</i> is specified, the -I option is assumed. If more than one <i>termname</i> is specified, the -d option is assumed.
OPTIONS	<p>The -d, -c, and -n options can be used for comparisons. infocmp compares the terminfo description of the first terminal <i>termname</i> with each of the descriptions given by the entries for the other terminal's <i>termname</i>. If a capability is defined for only one of the terminals, the value returned will depend on the type of the capability: F for boolean variables, -1 for integer variables, and NULL for string variables.</p> <p>-d Produce a list of each capability that is different between two entries. This option is useful to show the difference between two entries, created by different people, for the same or similar terminals.</p> <p>-c Produce a list of each capability that is common between two entries. Capabilities that are not set are ignored. This option can be used as a quick check to see if the -u option is worth using.</p> <p>-n Produce a list of each capability that is in neither entry. If no <i>termname</i> is given, the environment variable TERM will be used for both of the <i>termnames</i>. This can be used as a quick check to see if anything was left out of a description.</p> <p>The -I, -L, and -C options will produce a source listing for each terminal named.</p> <p>-I Use the terminfo names.</p> <p>-L Use the long C variable name listed in <code>< term.h ></code>.</p> <p>-C Use the termcap names. The source produced by the -C option may be used directly as a termcap entry, but not all of the parameterized strings may be changed to the termcap format. infocmp will attempt to convert most of the parameterized information, but anything not converted will be plainly marked in the output and commented out. These should be edited by hand.</p> <p>-r When using -C, put out all capabilities in termcap form.</p> <p>If no <i>termname</i> is given, the environment variable TERM will be used for the terminal name.</p> <p>All padding information for strings will be collected together and placed at the beginning of the string where termcap expects it. Mandatory padding (padding information with a trailing '/') will become optional.</p>

All **termcap** variables no longer supported by **terminfo**, but are derivable from other **terminfo** variables, will be displayed. Not all **terminfo** capabilities will be translated; only those variables which were part of **termcap** will normally be displayed. Specifying the **-r** option will take off this restriction, allowing all capabilities to be displayed in **termcap** form.

Note that because padding is collected to the beginning of the capability, not all capabilities are displayed. Mandatory padding is not supported. Because **termcap** strings are not as flexible, it is not always possible to convert a **terminfo** string capability into an equivalent **termcap** format. A subsequent conversion of the **termcap** file back into **terminfo** format will not necessarily reproduce the original **terminfo** source.

Some common **terminfo** parameter sequences, their **termcap** equivalents, and some terminal types which commonly have such sequences, are:

terminfo	termcap	Representative Terminals
%p1%c	%.	adm
%p1%d	%d	hp, ANSI standard, vt100
%p1%'x'%+%c	%+x	concept
%i	%i	ANSI standard, vt100
%p1%?'x'%>%t%p1%'y'%+%;	%>xy	concept
%p2 is printed before %p1	%r	hp

-u Produce a **terminfo** source description of the first terminal *termname* which is relative to the sum of the descriptions given by the entries for the other terminals' *termnames*. It does this by analyzing the differences between the first *termname* and the other *termnames* and producing a description with **use=** fields for the other terminals. In this manner, it is possible to retrofit generic **terminfo** entries into a terminal's description. Or, if two similar terminals exist, but were coded at different times, or by different people so that each description is a full description, using **infocmp** will show what can be done to change one description to be relative to the other.

A capability is displayed with an at-sign (@) if it no longer exists in the first *termname*, but one of the other *termname* entries contains a value for it. A capability's value is displayed if the value in the first *termname* is not found in any of the other *termname* entries, or if the first of the other *termname* entries that has this capability gives a different value for that capability.

The order of the other *termname* entries is significant. Since the **terminfo** compiler **tic** does a left-to-right scan of the capabilities, specifying two **use=** entries that contain differing entries for the same capabilities will produce different results, depending on the order in which the entries are given. **infocmp** will flag any such inconsistencies between the other *termname* entries as they are found.

Alternatively, specifying a capability *after* a **use=** entry that contains, it will cause the second specification to be ignored. Using **infocmp** to recreate a description can be a useful check to make sure that everything was specified correctly in the original source description.

Another error that does not cause incorrect compiled files, but will slow down the compilation time, is specifying superfluous **use=** fields. **infocmp** will flag any superfluous **use=** fields.

- s** Sorts the fields within each type according to the argument below:
 - d** Leave fields in the order that they are stored in the **terminfo** database.
 - i** Sort by **terminfo** name.
 - l** Sort by the long C variable name.
 - c** Sort by the **termcap** name.

If the **-s** option is not given, the fields are sorted alphabetically by the **terminfo** name within each type, except in the case of the **-C** or the **-L** options, which cause the sorting to be done by the **termcap** name or the long C variable name, respectively.

- v** Print out tracing information on standard error as the program runs.
- V** Print out the version of the program in use on standard error and exit.
- 1** Print the fields one to a line. Otherwise, the fields are printed several to a line to a maximum width of 60 characters.
- w width** Changes the output to *width* characters.

The location of the compiled **terminfo** database is taken from the environment variable **TERMINFO**. If the variable is not defined, or the terminal is not found in that location, the system **terminfo** database, usually in **/usr/share/lib/terminfo**, is used. The options **-A** and **-B** may be used to override this location.

- A directory** Set **TERMINFO** for the first *termname*.
- B directory** Set **TERMINFO** for the other *termnames*. With this, it is possible to compare descriptions for a terminal with the same name located in two different databases. This is useful for comparing descriptions for the same terminal created by different people.

FILES

/usr/share/lib/terminfo/?/* Compiled terminal description database.

SEE ALSO

captainfo(1M), **tic(1M)**, **curses(3X)**, **terminfo(4)**

NAME	init, telinit – process control initialization								
SYNOPSIS	<code>/sbin/init [0123456abcQqSs]</code> <code>/etc/telinit [0123456abcQqSs]</code>								
AVAILABILITY	SUNWcsu								
DESCRIPTION	init is a general process spawner. Its primary role is to create processes from information stored in the file <code>/etc/inittab</code> .								
Run Level Defined	At any given time, the system is in one of eight possible run levels. A run level is a software configuration under which only a selected group of processes exists. Processes spawned by init for each of these run levels are defined in <code>/etc/inittab</code> . init can be in one of eight run levels, 0–6 and S or s (S and s are identical). The run level changes when a privileged user runs <code>/sbin/init</code> . This sends appropriate signals to the original init spawned by the operating system at boot time, saying which run level to invoke.								
init and System Booting	<p>When the system is booted, init is invoked and the following occurs. First, it reads <code>/etc/default/init</code> to set environment variables. This is typically where TZ (time zone) and locale-related environments such as LANG or LC_CTYPE get set.</p> <p>init then looks in <code>/etc/inittab</code> for the initdefault entry (see <code>inittab(4)</code>). If the initdefault entry:</p> <table border="0" style="margin-left: 2em;"> <tr> <td style="padding-right: 1em;">exists</td> <td>init usually uses the run level specified in that entry as the initial run level to enter.</td> </tr> <tr> <td style="padding-right: 1em;">does not exist</td> <td><code>/etc/inittab</code>, init asks the user to enter a run level from the system console.</td> </tr> <tr> <td style="padding-right: 1em;">S or s</td> <td>init goes to the single-user state. In this state, the system console device (<code>/dev/console</code>) is opened for reading and writing and the command <code>/sbin/su</code>, (see <code>su(1M)</code>), is invoked. Use either init or telinit to change the run level of the system. Note that if the shell is terminated (using an end-of-file), init only re-initializes to the single-user state if <code>/etc/inittab</code> does not exist.</td> </tr> <tr> <td style="padding-right: 1em;">0-6</td> <td>init enters the corresponding run level. Run levels 0, 5, and 6 are reserved states for shutting the system down. Run levels 2, 3, and 4 are available as multi-user operating states.</td> </tr> </table> <p>If this is the first time since power up that init has entered a run level other than single-user state, init first scans <code>/etc/inittab</code> for boot and bootwait entries (see <code>inittab(4)</code>). These entries are performed before any other processing of <code>/etc/inittab</code> takes place, providing that the run level entered matches that of the entry. In this way any special initialization of the operating system, such as mounting file systems, can take place before users are allowed onto the system. init then scans <code>/etc/inittab</code> and executes all other entries that</p>	exists	init usually uses the run level specified in that entry as the initial run level to enter.	does not exist	<code>/etc/inittab</code> , init asks the user to enter a run level from the system console.	S or s	init goes to the single-user state. In this state, the system console device (<code>/dev/console</code>) is opened for reading and writing and the command <code>/sbin/su</code> , (see <code>su(1M)</code>), is invoked. Use either init or telinit to change the run level of the system. Note that if the shell is terminated (using an end-of-file), init only re-initializes to the single-user state if <code>/etc/inittab</code> does not exist.	0-6	init enters the corresponding run level. Run levels 0 , 5 , and 6 are reserved states for shutting the system down. Run levels 2 , 3 , and 4 are available as multi-user operating states.
exists	init usually uses the run level specified in that entry as the initial run level to enter.								
does not exist	<code>/etc/inittab</code> , init asks the user to enter a run level from the system console.								
S or s	init goes to the single-user state. In this state, the system console device (<code>/dev/console</code>) is opened for reading and writing and the command <code>/sbin/su</code> , (see <code>su(1M)</code>), is invoked. Use either init or telinit to change the run level of the system. Note that if the shell is terminated (using an end-of-file), init only re-initializes to the single-user state if <code>/etc/inittab</code> does not exist.								
0-6	init enters the corresponding run level. Run levels 0 , 5 , and 6 are reserved states for shutting the system down. Run levels 2 , 3 , and 4 are available as multi-user operating states.								

are to be processed for that run level.

To spawn each process in **/etc/inittab**, **init** reads each entry and for each entry that should be respawned, it forks a child process. After it has spawned all of the processes specified by **/etc/inittab**, **init** waits for one of its descendant processes to die, a **powerfail** signal, or a signal from another **init** or **telinit** process to change the system's run level. When one of these conditions occurs, **init** re-examines **/etc/inittab**.

inittab Additions

New entries can be added to **/etc/inittab** at any time; however, **init** still waits for one of the above three conditions to occur before re-examining **/etc/inittab**. To get around this, **init Q** or **init q** command wakes **init** to re-examine **/etc/inittab** immediately.

When **init** comes up at boot time and whenever the system changes from the single-user state to another run state, **init** sets the **ioctl(2)** states of the console to those modes saved in the file **/etc/ioctl.syscon**. **init** writes this file whenever the single-user state is entered.

Run Level Changes

When a run level change request is made, **init** sends the warning signal (**SIGTERM**) to all processes that are undefined in the target run level. **init** waits five seconds before forcibly terminating these processes by sending a kill signal (**SIGKILL**).

When **init** receives a signal telling it that a process it spawned has died, it records the fact and the reason it died in **/var/adm/utmp** and **/var/adm/wtmp** if it exists (see **who(1)**). A history of the processes spawned is kept in **/var/adm/wtmp**.

If **init** receives a **powerfail** signal (**SIGPWR**) it scans **/etc/inittab** for special entries of the type **powerfail** and **powerwait**. These entries are invoked (if the run levels permit) before any further processing takes place. In this way **init** can perform various cleanup and recording functions during the powerdown of the operating system.

/etc/defaults/init File

Default values can be set for the following flags in **/etc/default/init**. For example:
TZ=US/Pacific

TZ	Either specifies the timezone information (see ctime(3C)) or the name of a timezone information file /usr/share/lib/zoneinfo .
LC_CTYPE	Character characterization information.
LC_MESSAGES	Message translation.
LC_MONETARY	Monetary formatting information.
LC_NUMERIC	Numeric formatting information.
LC_TIME	Time formatting information.
LC_ALL	If set, all other LC_* environmental variables take-on this value.
LANG	If LC_ALL is not set, and any particular LC_* is also not set, the value of LANG is used for that particular environmental variable.

telinit

telinit, which is linked to **/sbin/init**, is used to direct the actions of **init**. It takes a one-character argument and signals **init** to take the appropriate action.

OPTIONS

- 0** Go into firmware.
- 1** Put the system in system administrator mode. All file systems are mounted. Only a small set of essential kernel processes are left running. This mode is for administrative tasks such as installing optional utility packages. All files are accessible and no users are logged in on the system.
- 2** Put the system in multi-user mode. All multi-user environment terminal processes and daemons are spawned. This state is commonly referred to as the multi-user state.
- 3** Extend multi-user mode by making local resources available over the network.
- 4** Is available to be defined as an alternative multi-user environment configuration. It is not necessary for system operation and is usually not used.
- 5** Shut the machine down so that it is safe to remove the power. Have the machine remove power, if possible.
- 6** Stop the operating system and reboot to the state defined by the **initdefault** entry in **/etc/inittab**.
- a, b, c** process only those **/etc/inittab** entries having the **a**, **b**, or **c** run level set. These are pseudo-states, which may be defined to run certain commands, but which do not cause the current run level to change.
- Q, q** Re-examine **/etc/inittab**.
- S, s** Enter single-user mode. When this occurs, the terminal that executed this command becomes the system console. This is the only run level that doesn't require the existence of a properly formatted **/etc/inittab** file. If this file does not exist, then by default, the only legal run level that **init** can enter is the single-user mode. When the system comes up to **S** or **s**, file systems for users' files are not mounted and only essential kernel processes are running. When the system comes down to **S** or **s**, all mounted file systems remain mounted, and all processes started by **init** that should only be running in multi-user mode are killed. In addition, any process that has a **utmp** entry will be killed. This last condition insures that all port monitors started by the SAC are killed and all services started by these port monitors, including **ttymon** login services, are killed. Other processes not started directly by **init** will remain running. For example, **cron** remains running.

FILES

/etc/inittab	controls process dispatching by init
/var/adm/utmp	accounting information
/var/adm/wtmp	history of all logins since file was last created
/etc/ioctl.syscon	
/dev/console	system console device
/etc/default/init	environment variables.

SEE ALSO

login(1), **sh(1)**, **stty(1)**, **who(1)**, **shutdown(1M)**, **su(1M)**, **ttymon(1M)**, **ioctl(2)**, **kill(2)**, **ctime(3C)**, **inittab(4)**, **utmp(4)**, **utmpx(4)**, **termio(7I)**

DIAGNOSTICS

If **init** finds that it is respawning an entry from **/etc/inittab** more than ten times in two minutes, assumes that there is an error in the command string in the entry, and generates an error message on the system console. It will then refuse to respawn this entry until either five minutes has elapsed or it receives a signal from a user-spawned **init** or **telinit**. This prevents **init** from eating up system resources when someone makes a typographical error in the **inittab** file, or a program is removed that is referenced in **/etc/inittab**.

NOTES

init and **telinit** can be run only by a privileged user.

The **S** or **s** state must not be used indiscriminately in **/etc/inittab**. When modifying this file, it is best to avoid adding this state to any line other than **initdefault**.

If a default state is not specified in the **initdefault** entry in **/etc/inittab**, state **6** is entered. Consequently, the system will loop by going to firmware and rebooting continuously.

If the **utmp** file cannot be created when booting the system, the system will boot to state “**s**” regardless of the state specified in the **initdefault** entry in **/etc/inittab**. This can occur if the **/var** file system is not accessible.

NAME	install – install commands
SYNOPSIS	<pre> /usr/sbin/install -c <i>dira</i> [-m <i>mode</i>] [-u <i>user</i>] [-g <i>group</i>] [-o] [-s] <i>file</i> /usr/sbin/install -f <i>dirb</i> [-m <i>mode</i>] [-u <i>user</i>] [-g <i>group</i>] [-o] [-s] <i>file</i> /usr/sbin/install -n <i>dirc</i> [-m <i>mode</i>] [-u <i>user</i>] [-g <i>group</i>] [-o] [-s] <i>file</i> /usr/sbin/install -d -i [-m <i>mode</i>] [-u <i>user</i>] [-g <i>group</i>] [-o] [-s] <i>dirx</i> ... /usr/sbin/install [-m <i>mode</i>] [-u <i>user</i>] [-g <i>group</i>] [-o] [-s] <i>file</i> [<i>dirx</i> ...] </pre>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>install is most commonly used in “makefiles” (see make(1S)) to install a <i>file</i> in specific locations, or to create directories within a file system. Each <i>file</i> is installed by copying it into the appropriate directory.</p> <p>install uses no special privileges to copy files from one place to another. The implications of this are:</p> <ul style="list-style-type: none"> • You must have permission to read the files to be installed. • You must have permission to copy into the destination directory. • You must have permission to change the modes on the final copy of the file if you want to use the -m option. • You must be super-user if you want to specify the ownership of the installed file with the -u or -g options. If you are not the super-user, the installed file will be owned by you, regardless of who owns the original. <p>install prints messages telling the user exactly what files it is replacing or creating and where they are going.</p> <p>If no options or directories (<i>dirx</i> ...) are given, install searches a set of default directories (/bin, /usr/bin, /etc, /lib, and /usr/lib, in that order) for a file with the same name as <i>file</i>. When the first occurrence is found, install issues a message saying that it is overwriting that file with <i>file</i>, and proceeds to do so. If the file is not found, the program states this and exits.</p> <p>If one or more directories (<i>dirx</i> ...) are specified after <i>file</i>, those directories are searched before the default directories.</p>
OPTIONS	<p>-c <i>dira</i> Install <i>file</i> in the directory specified by <i>dira</i>, if <i>file</i> does not yet exist. If it is found, install issues a message saying that the file already exists, and exits without overwriting it.</p> <p>-f <i>dirb</i> Force <i>file</i> to be installed in given directory, even if the file already exists. If the file being installed does not already exist, the mode and owner of the new file will be set to 755 and bin, respectively. If the file already exists, the mode and owner will be that of the already existing file.</p> <p>-n <i>dirc</i> If <i>file</i> is not found in any of the searched directories, it is put in the directory specified in <i>dirc</i>. The mode and owner of the new file will be set to 755 and bin, respectively.</p> <p>-d Create a directory. Missing parent directories are created as required as in</p>

- mkdir -p.** If the directory already exists, the owner, group and mode will be set to the values given on the command line.
- i** Ignore default directory list, searching only through the given directories (*dirx ...*).
 - m mode** The mode of the new file is set to *mode*. Set to **0755** by default.
 - u user** The owner of the new file is set to *user*. Only available to the super-user. Set to **bin** by default.
 - g group** The group id of the new file is set to *group*. Only available to the super-user. Set to **bin** by default.
 - o** If *file* is found, save the “found” file by copying it to **OLDfile** in the directory in which it was found. This option is useful when installing a frequently used file such as **/bin/sh** or **/lib/saf/ttymon**, where the existing file cannot be removed.
 - s** Suppress printing of messages other than error messages.

SEE ALSO **chgrp(1), chmod(1), chown(1), cp(1), make(1S), mkdir(1), chown(1M)**

NAME	install4x – install SunOS 4.x diskless/dataless client support on Solaris 2.x servers
SYNOPSIS	/usr/sbin/install4x [-e <i>export_root</i>] [-m <i>cdrom_mount_point</i>]
AVAILABILITY	SUNWhinst
DESCRIPTION	<p>install4x is most commonly used after upgrading a SunOS 4.x server to Solaris 2.x. It is used to install the components of a SunOS 4.x system that are required to support diskless/dataless clients that existed before the converting.</p> <p>The interface to install4x is interactive. If the user does not specify options on the command line, they are prompted for any required options.</p>
OPTIONS	<p>-e <i>export_root</i> Specify the location of the <i>export</i> directory, where install4x installs the SunOS 4.x components. If unspecified, the default value is /export.</p> <p>-m <i>cdrom_mount_point</i> Specify the location of the CD-ROM containing the SunOS 4.x distribution. If unspecified, the default value is /cdrom.</p>
SEE ALSO	<p>convert4x(1M), discover4x(1M) <i>SPARC: Installing Solaris Software</i> <i>x86: Installing Solaris Software</i></p>

NAME	install_scripts, add_install_client, rm_install_client, setup_install_server, check – scripts used to install the Solaris software
SYNOPSIS	<pre>cdrom-mnt-pt/add_install_client [-i IP_address] [-e Ethernet_address] [-s server_name:path] [-c server_name:path] [-n [server]:name_service [(netmask)]] host_name platform_name cdrom-mnt-pt/rm_install_client host_name cdrom-mnt-pt/setup_install_server [-b] install_dir_path cdrom-mnt-pt/check [-p] [-r rulesfile] install_dir_path</pre>
AVAILABILITY	SUNWcdrom (Solaris CD)
DESCRIPTION	<p>These commands are located on the Solaris CD in the /cdrom/cdrom0/s0 directory. (If the Solaris CD has been copied to a local disk, these scripts will be in the path to that directory.) You can use them for a variety of installation tasks. Specifically,</p> <ul style="list-style-type: none"> • Use add_install_client and rm_install_client to add or remove clients for network installation. • Use setup_install_server to copy the Solaris CD to a disk or to copy just the boot software of the Solaris CD to a disk (i.e., set up a boot server) • Use check to validate the rules in a rules file (this is only necessary if you are setting up a custom JumpStart installation).
OPTIONS	The following options are supported:
add_install_client	<p>-i IP_address Specify the IP address of the client to be installed.</p> <p>-e Ethernet_address Specify the Ethernet address of the system to be installed.</p> <p>-s server_name:path This option is required only when using add_install_client from a boot server. Specify the name of the server and the absolute path of the Solaris installation image that will be used for this installation. <i>path</i> is either the path to a mounted Solaris CD or a path to a directory with a copy of the Solaris CD.</p> <p>-c server_name:path This option is required only to specify a JumpStart directory for a custom JumpStart installation. <i>server_name</i> is the host name of the server with a JumpStart directory. <i>path</i> is the absolute path to the JumpStart directory.</p> <p>-n [server]:name_service This option specifies which name service should be used during system configuration. This sets the 'ns' bootparams(4) keyword. Valid entries are 'nis', 'nisplus', and 'none'. <i>server</i> is the name of the server or IP address of the specified name service. If the server specified is on a different subnet, then the netmask may be needed to enable to client to contact the server. <i>netmask</i> is a series of four</p>

numbers separated by periods, specifying which portion of an IP address is the network part, and which is the host part.

- rm_install_client** **-b** This option sets up the server only as a boot server.
- check** **-p** *install_dir_path* Specifies the absolute path to the Solaris installation image (either the mounted Solaris CD-ROM or a copy of the Solaris CD-ROM on the local disk). Using this option ensures you are using the most recent check program to validate your **rules** file.
- r** *rulesfile* Specifies a rules file other than the one named **rules**. Using this option, you can test the validity of a rule before integrating it into the rules file. **check** will report whether or not the rule is valid, but it will not create the **rules.ok** file necessary for a custom JumpStart installation.

OPERANDS

The following operands are supported:

- add_install_client** *host_name* This is the name of the client that you are going to install.
- platform_name* Vendor-defined grouping of hardware platforms for the purpose of distributing specific software. Examples of valid platform names are:
- | System | Platform Name |
|-----------------|---------------|
| x86 | i86pc |
| SPARCstation 1+ | sun4c |
| SPARCstation LX | sun4c |
- rm_install_client** *host_name* This is the name of the client that you are going to remove.
- platform_name* Vendor-defined grouping of hardware platforms for the purpose of distributing specific software. (See previous list for valid arguments.)
- setup_install_server** *install_dir_path* Specify the absolute path of the directory in which you want to copy the Solaris software.
- platform_name* Vendor-defined grouping of hardware platforms for the purpose of distributing

EXAMPLES

The following **add_install_client** commands add clients for network installation from a mounted Solaris CD on an install server.

example# cd /cdrom/cdrom0/s0

example# ./add_install_client system_1 sun4c

example# ./add_install_client system_2 i86pc

The following **add_install_client** commands add clients for network installation from a mounted Solaris CD on an install server. The **-c** option specifies a server and path to a JumpStart directory that has a rules and profile files for performing a custom JumpStart installation.

```
example# cd /cdrom/cdrom0/s0
```

```
example# ./add_install_client -c install_server:/jumpstart system_1 sun4c
```

```
example# ./add_install_client -c install_server:/jumpstart system_2 i86pc
```

The following **rm_install_client** commands remove system information about the named clients on the install server.

```
example# cd /cdrom/cdrom0/s0
```

```
example# ./rm_install_client holmes
```

```
example# ./rm_install_client watson
```

The following **setup_install_server** command copies the mounted Solaris CD to a directory named **/export/install** on the local disk. (This requires approximately 200 Mbytes of disk space.)

```
example# cd /cdrom/cdrom0/s0
```

```
example# ./setup_install_server /export/install
```

The following **setup_install_server** command copies the boot software of a mounted Solaris CD to a directory named **/boot_dir** on system that is going to be a boot server for a subnet. You must enter the command once for each client architecture to be installed on the subnet.

```
example# cd /cdrom/cdrom0/s0
```

```
example# ./setup_install_server -b
```

```
example# ./setup_install_server -b
```

The following **check** command validates the syntax of the rules file used for a custom JumpStart installation.

```
example# cd jumpstart_dir_path
```

```
example# ./check -p /cdrom/cdrom0/s0
```

EXIT STATUS

The following exit values are returned:

0	success
1	an error has occurred.

NAME	installboot – install bootblocks in a disk partition
SPARC SYNOPSIS	installboot <i>bootblk</i> <i>raw-disk-device</i>
x86 SYNOPSIS	installboot <i>pboot</i> <i>bootblk</i> <i>raw-disk-device</i>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>The boot(1M) program is loaded from disk by the bootblock program which resides in the boot area of a disk partition.</p> <p><i>bootblk</i> is the name of the bootblock code. <i>raw-disk-device</i> is the name of the disk device onto which the bootblock code is to be installed; it must be a character device which is readable and writable. Naming conventions for a SCSI or IPI drive are <i>c?t?d?s?</i> and <i>c?d?s?</i> for an IDE drive. <i>pboot</i> is the name of the partition boot file.</p> <p>The ufs boot objects are platform-dependent, and reside in the <i>/usr/platform/platform-name/lib/fs/ufs</i> directory. The platform name can be found using the -i option of uname(1).</p>
SPARC EXAMPLES	<p>To install a ufs bootblock on slice 0 of target 0 on controller 1, of the platform where the command is being run, use:</p> <pre>example# installboot /usr/platform/'uname -i'/lib/fs/ufs/bootblk \ /dev/rdisk/c1t0d0s0</pre>
x86 EXAMPLES	<p>To install the ufs bootblock and partition boot program on slice 2 of target 0 on controller 1, of the platform where the command is being run, use:</p> <pre>example# installboot /usr/platform/'uname -i'/lib/fs/ufs/pboot \ /usr/platform/'uname -i'/lib/fs/ufs/bootblk /dev/rdisk/c1t0d0s2</pre>
SEE ALSO	<p>od(1), uname(1), boot(1M), init(1M), kadb(1M), kernel(1M), reboot(1M), rpc.bootparamd(1M), init.d(4)</p> <p><i>SPARC: Installing Solaris Software</i> <i>x86: Installing Solaris Software</i></p>
SPARC only x86 only	<p>monitor(1M) fmthard(1M), fdisk(1M)</p>
WARNINGS	installboot will fail if the <i>bootblk</i> or <i>pboot</i> files don't exist or if the raw disk device isn't a character device.

NAME	installf – add a file to the software installation database
SYNOPSIS	installf [-c <i>class</i>] <i>pkginst</i> <i>pathname</i> [<i>ftype</i> [<i>major</i> <i>minor</i>] [<i>mode</i> <i>owner</i> <i>group</i>]] installf [-c <i>class</i>] <i>pkginst</i> – installf -f [-c <i>class</i>] <i>pkginst</i>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>installf informs the system that a <i>pathname</i> not listed in the pkgmap(4) file is being created or modified. It should be invoked before any file modifications have occurred.</p> <p>When the second synopsis is used, the <i>pathname</i> descriptions will be read from standard input. These descriptions are the same as would be given in the first synopsis but the information is given in the form of a list. The descriptions should be in the form: <i>pathname</i> [<i>ftype</i> [<i>major</i> <i>minor</i>] [<i>mode</i> <i>owner</i> <i>group</i>]].</p> <p>After all files have been appropriately created and/or modified, installf should be invoked with the -f synopsis to indicate that installation is final. Links will be created at this time and, if attribute information for a <i>pathname</i> was not specified during the original invocation of installf, or was not already stored on the system, the current attribute values for the <i>pathname</i> will be stored. Otherwise, installf verifies that attribute values match those given on the command line, making corrections as necessary. In all cases, the current content information is calculated and stored appropriately.</p>
OPTIONS	<p>-c <i>class</i> Class to which installed objects should be associated. Default class is none.</p> <p><i>pkginst</i> Name of package instance with which the <i>pathname</i> should be associated.</p> <p><i>pathname</i> Pathname that is being created or modified.</p> <p><i>ftype</i> A one-character field that indicates the file type. Possible file types include:</p> <ul style="list-style-type: none"> b block special device c character special device d directory e a file to be edited upon installation or removal f a standard executable or data file l linked file p named pipe s symbolic link v volatile file (one whose contents are expected to change) x an exclusive directory <p><i>major</i> The major device number. The field is only specified for block or character special devices.</p> <p><i>minor</i> The minor device number. The field is only specified for block or character special devices.</p>

- mode* The octal mode of the file (for example, 0664). A question mark (?) indicates that the mode will be left unchanged, implying that the file already exists on the target machine. This field is not used for linked or symbolically linked files.
- owner* The owner of the file (for example, **bin** or **root**). The field is limited to 14 characters in length. A question mark (?) indicates that the owner will be left unchanged, implying that the file already exists on the target machine. This field is not used for linked or symbolically linked files.
- group* The group to which the file belongs (for example, **bin** or **sys**). The field is limited to 14 characters in length. A question mark (?) indicates that the group will be left unchanged, implying that the file already exists on the target machine. This field is not used for linked or symbolically linked files.
- f** Indicates that installation is complete. This option is used with the final invocation of **installf** (for all files of a given class).

EXAMPLES

The following example shows the use of **installf**, invoked from an optional preinstall or postinstall script:

```
#create /dev/xt directory
#(needs to be done before drvinstall)
installf $PKGINST /dev/xt d 755 root sys | |
    exit 2
majno='/usr/sbin/drvinstall -m /etc/master.d/xt
    -d $BASEDIR/data/xt.o -v1.0' | |
    exit 2
i=00
while [ $i -lt $limit ]
do
    for j in 0 1 2 3 4 5 6 7
    do
        echo /dev/xt${i}$j c $majno `expr $i * 8 + $j`
        644 root sys |
        echo /dev/xt${i}$j=/dev/xt/${i}$j
    done
    i=`expr $i + 1`
    [ $i -le 9 ] && i="0$i" #add leading zero
done | installf $PKGINST - | | exit 2
# finalized installation, create links
installf -f $PKGINST | | exit 2
```

SEE ALSO

pkginfo(1), **pkgmk(1)**, **pkgparam(1)**, **pkgproto(1)**, **pkgtrans(1)**, **pkgadd(1M)**, **pkgask(1M)**, **pkgchk(1M)**, **pkgrm(1M)**, **removef(1M)**

NOTES

When *ftype* is specified, all applicable fields, as shown below, must be defined:

FTYPE	REQUIRED FIELDS
p, x, d, f, v, or e	mode owner group
c or b	major minor mode owner group

The **installf** command will create directories, named pipes and special devices on the original invocation. Links are created when **installf** is invoked with the **-f** option to indicate installation is complete.

Links should be specified as *path1=path2*. *path1* indicates the destination and *path2* indicates the source file.

Files installed with **installf** will be placed in the class **none**, unless a class is defined with the command. Subsequently, they will be removed when the associated package is deleted. If this file should not be deleted at the same time as the package, be certain to assign it to a class which is ignored at removal time. If special action is required for the file before removal, a class must be defined with the command and an appropriate class action script delivered with the package.

When classes are used, **installf** must be used in one of the following forms:

```
installf -c class1 ...
installf -f -c class1 ...
installf -c class2 ...
installf -f -c class2 ...
```

NAME	iostat – report I/O statistics
SYNOPSIS	<code>/usr/bin/iostat [-cdDItx] [-l n] [disk ...] [interval [count]]</code>
DESCRIPTION	<p>iostat iteratively reports terminal and disk I/O activity, as well as CPU utilization. The first line of output is for all time since boot; each subsequent line is for the prior interval only.</p> <p>To compute this information, the kernel maintains a number of counters. For each disk, the kernel counts reads, writes, bytes read, and bytes written. The kernel also takes hi-res time stamps at queue entry and exit points, which allows it to keep track of the residence time and cumulative residence-length product for each queue. Using these values, iostat produces highly accurate measures of throughput, utilization, queue lengths, transaction rates and service time. For terminals collectively, the kernel simply counts the number of input and output characters.</p> <p>For more general system statistics, use sar(1), sar(1M), or vmstat(1M).</p> <p>See <i>Solaris 1.x to Solaris 2.x Transition Guide</i> for device naming conventions for disks.</p>
OPTIONS	<p>iostat's activity class options default to tdc (terminal, disk, and CPU). If any activity class options are specified, the default is completely overridden. Therefore, if only -d is specified, neither terminal nor CPU statistics will be reported. The last disk option specified (-d, -D, or -x) is the only one that is used.</p> <p>-c Report the percentage of time the system has spent in user mode, in system mode, waiting for I/O, and idling.</p> <p>-d For each disk, report the number of kilobytes transferred per second, the number of transfers per second, and the average service time in milliseconds.</p> <p>-D For each disk, report the reads per second, writes per second, and percentage disk utilization.</p> <p>-I Report the counts in each interval, rather than rates (where applicable).</p> <p>-t Report the number of characters read and written to terminals per second.</p> <p>-x For each disk, report extended disk statistics. The output is in tabular form.</p> <p>-l n Limit the number of disks included in the report to <i>n</i>; the disk limit defaults to 4 for -d and -D, and unlimited for -x. Note: disks explicitly requested (see <i>disk</i> below) are not subject to this disk limit.</p> <p><i>disk</i> Explicitly specify the disks to be reported; in addition to any explicit disks, any active disks up to the disk limit (see -I above) will also be reported.</p> <p><i>count</i> Only print <i>count</i> reports.</p> <p><i>interval</i> Report once each <i>interval</i> seconds.</p>

EXAMPLES**example%** iostat -xtc 5 2

disk	r/s	w/s	Kr/s	Kw/s	wait	extended disk statistics				tty				cpu	
						actv	svc_t	%w	%b	tin	tout	us	sy	wt	id
sd0	6.2	0.0	21.5	0.0	0.0	0.1	24.1	0	15	0	84	4	94	2	0
sd1	1.8	0.0	14.3	0.0	0.0	0.1	41.6	0	7						
sd2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	0						
sd3	5.6	0.2	25.7	0.2	0.0	0.1	22.5	0	13						

disk	r/s	w/s	Kr/s	Kw/s	wait	extended disk statistics				tty				cpu	
						actv	svc_t	%w	%b	tin	tout	us	sy	wt	id
sd0	2.6	3.0	20.7	22.7	0.1	0.2	59.2	6	19	0	84	3	85	11	0
sd1	4.2	1.0	33.5	8.0	0.0	0.2	47.2	2	23						
sd2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	0						
sd3	10.2	1.6	51.4	12.8	0.1	0.3	31.2	3	31						

example%

The fields have the following meanings:

disk name of the disk
r/s reads per second
w/s writes per second
Kr/s kilobytes read per second
Kw/s kilobytes written per second
wait average number of transactions waiting for service (queue length)
actv average number of transactions actively being serviced (removed from the queue but not yet completed)
svc_t average service time, in milliseconds
%w percent of time there are transactions waiting for service (queue non-empty)
%b percent of time the disk is busy (transactions in progress)

SEE ALSO**sar(1), sar(1M), vmstat(1M)***System Administration Guide, Volume II*

NAME	kadb – a kernel debugger
SPARC SYNOPSIS	ok boot <i>device_specifier</i> kadb > b kadb [-d] [<i>boot-flags</i>]
x86 SYNOPSIS	> run kadb [-d] [<i>b-flags</i>]
AVAILABILITY	SUNWcar
DESCRIPTION	<p>kadb is an interactive debugger with a user interface similar to that of adb(1), the assembly language debugger. kadb must be loaded prior to the standalone program it is to debug. kadb runs with the same environment as the standalone program, so it shares many resources with that program. The debugger is cognizant of and able to control multiple processors, should they be present in a system.</p> <p>Unlike adb, kadb runs in the same supervisor virtual address space as the program being debugged — although it maintains a separate context. The debugger runs as a <i>co-process</i> that cannot be killed (no :k) or rerun (no :r). There is no signal control (no :i, :t, or :Si), although the keyboard facilities (CTRL-C, CTRL-S, and CTRL-Q) are simulated.</p> <p>In the case of the UNIX system kernel, the keyboard abort sequence will suspend kernel operations and break into the debugger. On a SPARC system, the keyboard abort sequence is L1-A for the console, and BREAK for a serial line. On an x86 system, the keyboard abort sequence is CTRL-ALT-D for the console. kadb gives the prompt:</p> <p style="text-align: center;">kadb[#]</p> <p>where # is the CPU it is currently executing on.</p> <p>Since the kernel is composed of the core image file (kernel/unix) and the set of loadable modules already in memory, kadb has the capability of debugging all of these by traversing special data structures. kadb makes use of this feature to allow it to reference any kernel data structure, even if it resides in a loadable module. kadb will set the '-d' flag by default so the program being debugged can tell it is being watched. If this flag is not given, kadb will load and immediately run the default kernel.</p> <p>On a SPARC system, <i>device-specifier</i> specifies the device to load from (see monitor(1M)).</p> <p>Most adb commands function in kadb as expected. As with 'adb -k', \$p works when debugging kernels. The verbs ? and / are equivalent in kadb, since there is only one address space in use.</p>
Additional Commands	<p>kadb has the following additional commands. (Note: for the general syntax of adb commands, see adb(1).)</p> <p>[Like :e in adb(1), but requires only one keystroke and no RETURN character.</p> <p>] Like :s in adb(1), but requires only one keystroke and no RETURN character.</p>

**Additional x86
Commands**

- :a** Sets a hardware access (read or write) breakpoint using the processor hardware facilities. The syntax and action for this is the same as **:b** in **adb(1)**, with the following exceptions:
- the breakpoint will trigger if any bytes from the breakpoint for length bytes are being accessed. See **\$l** below for setting the length of a hardware breakpoint.
 - breakpoints should be aligned for the length specified. Any address is valid for length 1. Addresses divisible by 2 should be used for length 2 (short). Addresses divisible by 4 should be used for length 4 (long).
 - for an access to be detected it must complete the instruction that caused the access. This means that this breakpoint will stop after the access not before.
 - there are a limited number of hardware breakpoint registers (4) and when set this will use one.
 - since this breakpoint does not modify memory locations this will work on locations that are not in core at the time the breakpoint is set.
- :w** Like **:a** above, but will only breakpoint on a write to the specified address.
- :p** Like **:a** above, but will only breakpoint when an instruction at the specified address is run. Note that the **\$l** operation has no effect on this type of breakpoint. This breakpoint will occur before the instruction is executed.
- \$l** Sets the default data length for an access or write breakpoint. The syntax for this command is **[length]\$l**. Length can be set to one for byte, two for short and 4 for long word accesses. If length is left off byte will be assumed. Once set this value will affect any newly set access or write breakpoints. But will not affect ones set before this operation.
- \$b** Has two additional columns that **adb(1)** does not have. The first is the **type** column which indicates soft for a normal breakpoint, access for an access hardware breakpoint, write for a write hardware breakpoint, and inst for an instruction hardware breakpoint. The second is the **len** column which for access and write breakpoints indicate the length of the operation to break on.
- :i** Will input a byte for display from the address specified io port. For example **.B 330:i** will input from address port **330**.
- :o** Will output a byte to the address specified io port. **[address],[data]:o** will output data byte to address io port. For example **330,80:o** will output 80 to address port **330**.

OPTIONS	<p>-d Interactive startup. If this flag is given, kadb will stop after it is loaded and display the prompt as kadb: and the name of the default program to debug. For example:</p> <p>kadb: kernel/unix</p> <p>The user may either press Return to use the default, or backspace and type in the name of another program to debug.</p> <p><i>boot-flags</i></p> <p>You can specify boot flags as arguments when invoking kadb. Boot flags are passed to the program being debugged. See the manual page boot(1M) for relevant boot flags.</p>
USAGE Kernel Macros	<p>As with adb, macros can be used with kadb, but they cannot be read from a file at run-time. The kadb command \$M lists all the macros built into kadb.</p>
FILES	<p>/platform/platform-name/kadb the debugger itself</p> <p>/platform/platform-name/kernel/unix the default kernel</p>
SEE ALSO SPARC Only	<p>adb(1), boot(1M) kernel(1M), monitor(1M), obpsym(1M)</p>
DIAGNOSTICS	<p>kadb gives the same cryptic response as adb.</p>
WARNINGS	<p>SPARC: kadb cannot reliably single-step over instructions which change the processor status register.</p>
NOTES	<p><i>platform-name</i> can be found using the -i option of uname(1).</p>

NAME	kdmconfig – configure or unconfigure keyboard, display, and mouse options for OpenWindows and internationalization
SYNOPSIS	kdmconfig <i>-c</i> <i>-u</i> [<i>-s host</i>] [<i>-v</i>] [<i>-f</i>]
AVAILABILITY	x86 SUNWos86r
DESCRIPTION	<p>The kdmconfig program configures or unconfigures the /etc/defaultkb and /etc/openwin/server/etc/OWconfig files with the keyboard, display and mouse information relevant to a client's machine. It can also be used to set up the <i>display</i>, <i>pointer</i>, and <i>keyboard</i> entries in the bootparams database on a server machine.</p> <p>The kdmconfig program is not normally run from a shell prompt, except when run with the <i>-s</i> option. When configuring a client during an initial installation or reconfigure reboot, the sysidconfig(1M) program will invoke kdmconfig with the <i>-c</i> option, and when the user executes the sys-unconfig(1M) program, kdmconfig will be executed with the <i>-u</i> option.</p>
OPTIONS	<p>The valid options are:</p> <p><i>-c</i> Run the program in the configuration mode. This mode is used to create or update the defaultkb and OWconfig files. When invoked in this way, kdmconfig first looks for the relevant configuration information in the bootparams databases. The bootparams databases available to the client are all of the /etc/bootparams files on servers on the same subnet as the client, provided the server machine is running the bootparamd(1M) daemon. kdmconfig is invoked with the <i>-c</i> option when called by sysidconfig.</p> <p><i>-u</i> Unconfigure the system, returning it to an "out-of-the-box" state. In this state, the factory default keyboard, mouse and display are selected as a result of removing the /etc/openwin/server/etc/OWconfig and /etc/defaultkb files.</p> <p><i>-s client</i> Set up the bootparams database on this machine for the specified client. This option presents the same screens as it does when run on a client, but instead writes the resulting information to the /etc/bootparams file. Either the <i>-c</i> or the <i>-u</i> option must be used in conjunction with the <i>-s</i> option. Also, <i>-s</i> implies the use of the <i>-f</i> option. That is, the program will always present the screens to the user when invoked this way. This option will reconfigure the nsswitch.conf file to look for a bootparams database on a local server. This option is only available to the superuser.</p> <p><i>-v</i> Enable verbose mode. Normally, kdmconfig will not produce any output. This option is helpful for debugging, as it records the different actions taken by kdmconfig on stderr.</p>

-f Force screens mode. When invoked with **-f**, no network probing will be performed. This is helpful when debugging the client's configuration environment. Note that the **-s** option implies the use of **-f**, bypassing network probing when setting up a server.

FILES **/etc/defaultkb**
/etc/openwin/server/etc/OWconfig
/etc/bootparams
/etc/nsswitch.conf

SEE ALSO **admintool(1M)**, **bootparamd(1M)**, **sys-unconfig(1M)**, **sysidconfig(1M)**, **bootparams(4)**

NOTES The **-s** option is an uncommitted option, and may be replaced by functionality in the **admintool(1M)** hostmanager functions in a later release of Solaris.

NAME	kerbd – generates and validates Kerberos tickets for kernel RPC
SYNOPSIS	/usr/sbin/kerbd [-dg]
AVAILABILITY	SUNWcsu
DESCRIPTION	kerbd is the usermode daemon which interfaces between kernel RPC and the Kerberos key distribution center (KDC) for the purposes of generating and validating Kerberos authentication tickets. In addition, kerbd maps Kerberos user names into local user and group ids. By default, all groups that the requested user belongs to will be included in the grouplist credential. kerbd is automatically started when the system enters the multi-user state.
OPTIONS	-d Run in debug mode. kerbd will output various information about Kerberos tickets being processed. -g Do not initialize the grouplist in the user credential when mapped from Kerberos' principal name. If this option is selected, only each user's group from the passwd entry will be included in mapped credentials.
SEE ALSO	kdestroy(1) , kerberos(1) , kinit(1) , krb.conf(4)

NAME	kernel – UNIX system executable file containing basic operating system services
SYNOPSIS	/platform/platform-name/kernel/unix [-arsv] /kernel/genunix
AVAILABILITY	SUNWcar
DESCRIPTION	<p>The operating system image, or kernel, is the collection of software made up of the core image file (unix) and all of the modules loaded at any instant in time. The system will not function without a kernel to control it.</p> <p>The kernel is loaded by the boot(1M) command in a machine-specific way. The kernel may be loaded from disk or CDROM ("diskfull boot") or over the network ("diskless boot"). In either case, the directories under /platform and /kernel must be readable and must contain executable code which is able to perform the required kernel service. If the -a flag is given, the user is able to supply different pathnames for the default locations of the kernel and modules.</p> <p>The 'moddir' variable contains a colon-separated list of directories that the kernel searches for modules. 'moddir' can be set in the /etc/system file; it defaults to /platform/platform-name/kernel:/kernel:/usr/kernel (where <i>platform-name</i> is the name of the platform implementation).</p> <p><i>platform-name</i> can be found using the -i option of uname(1).</p> <p>The kernel configuration can be controlled using the /etc/system file (see system(4)). /kernel/genunix is the platform-independent component of the base kernel.</p>
OPTIONS	<p>-a Ask the user for configuration information, such as where to find the system file, where to mount root, and even override the name of the kernel itself. Default responses will be contained in square brackets ([]), and the user may simply enter RETURN to use the default response (note that RETURN is labeled ENTER on some keyboards). /dev/null may be entered at the prompt which asks for the pathname of the system configuration file. See system(4).</p> <p>-r Reconfiguration boot. The system will probe all attached hardware devices and assign nodes in the file system to represent only those devices actually found. It will also configure the logical namespace in /dev as well as the physical namespace in /devices. See add_drv(1M) and rem_drv(1M) for additional information about maintaining device drivers.</p> <p>-s Boot only to init level 's'. See init(1M).</p> <p>-v Boot with verbose messages enabled. If this flag is not given, the messages are still printed, but the output is directed to the system logfile. See syslogd(1M).</p>
SPARC EXAMPLES	<p>To boot in single-user interactive mode, use one of the following:</p> <p>ok boot -as</p> <p>ok boot kernel/unix -as</p>

ok boot disk3 kernel/unix -as**x86 EXAMPLES**

To boot in single-user interactive mode, use one of the following:

>**b -as**

>**b kernel/unix -as**

FILES

/kernel Contains kernel components common to all platforms within a particular instruction set that are needed for booting the system. of the core image file.

/platform/platform-name/kernel

The platform-specific kernel components.

/usr/kernel Contains kernel components common to all platforms within a particular instruction set.

Each of the directories listed above can potentially contain the following subdirectories:

drv Loadable device drivers
exec The modules to execute processes
fs File system modules
misc Miscellaneous system-related modules
sched Operating system schedulers
strmod System V STREAMS loadable modules
sys Loadable system calls

x86 Only

mach x86 hardware support

SEE ALSO

uname(1), **add_drv(1M)**, **boot(1M)**, **kadb(1M)**, **rem_drv(1M)**, **savecore(1M)**, **syslogd(1M)**, **system(4)**

SPARC Only

monitor(1M)

DIAGNOSTICS

The kernel gives various warnings and error messages. If the kernel detects an unrecoverable fault, it will panic or halt.

BUGS

Bugs in the kernel often result in kernel panics.

Reconfiguration boot does not currently remove filesystem entries for devices that have been physically removed from the system.

NAME	keyserv – server for storing private encryption keys
SYNOPSIS	keyserv [-d] [-D] [-n]
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>keyserv is a daemon that is used for storing the private encryption keys of each user logged into the system. These encryption keys are used for accessing secure network services such as secure NFS and NIS+.</p> <p>Normally, root's key is read from the file /etc.rootkey when the daemon is started. This is useful during power-fail reboots when no one is around to type a password.</p>
OPTIONS	<p>-d Disable the use of default keys for nobody.</p> <p>-D Run in debugging mode and log all requests to keyserv.</p> <p>-n Root's secret key is not read from /etc.rootkey. Instead, keyserv prompts the user for the password to decrypt root's key stored in the publickey database and then stores the decrypted key in /etc.rootkey for future use. This option is useful if the /etc.rootkey file ever gets out of date or corrupted.</p>
FILES	/etc.rootkey
SEE ALSO	keylogin(1) , keylogout(1) , publickey(4)

NAME	killall – kill all active processes
SYNOPSIS	<code>/usr/sbin/killall [<i>signal</i>]</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>killall is used by shutdown(1M) to kill all active processes not directly related to the shutdown procedure.</p> <p>killall terminates all processes with open files so that the mounted file systems will be unbusied and can be unmounted.</p> <p>killall sends <i>signal</i> (see kill(1)) to the active processes. If no <i>signal</i> is specified, a default of 15 is used.</p> <p>The killall command can be run only by the super-user.</p>
SEE ALSO	kill(1) , ps(1) , fuser(1M) , shutdown(1M) , signal(3C)

NAME	labelit – list or provide labels for file systems
SYNOPSIS	labelit [-F <i>FSType</i>] [-V] [-o <i>specific_options</i>] <i>special</i> [<i>operands</i>]
AVAILABILITY	SUNWcsu
DESCRIPTION	labelit is used to write or display labels on unmounted disk file systems.
OPTIONS	<p>The following options are supported:</p> <p>-F <i>FSType</i> Specify the <i>FSType</i> on which to operate. The <i>FSType</i> should either be specified here or be determinable from /etc/vfstab by matching the <i>special</i> with an entry in the table. If no matching entry is found, the default file system type specified in /etc/default/fs will be used.</p> <p>-V Echo complete command line. This option may be used to verify and validate the command line. Additional information obtained using a /etc/vfstab lookup is included in the output. The command is not executed.</p> <p>-o <i>FSType-specific_options</i> Specify <i>FSType</i>-specific options in a comma separated (without spaces) list of suboptions and keyword-attribute pairs for interpretation by the <i>FSType</i>-specific module of the command.</p>
OPERANDS	<p>The following operands are supported:</p> <p><i>special</i> name should be the disk partition (for example, /dev/rdisk/c0d0s6). The device may not be on a remote machine.</p> <p><i>operands</i> <i>FSType</i>-specific operands. Consult the manual page of the <i>FSType</i>-specific labelit(1M) command for detailed descriptions. If no operands are specified, labelit will display the value of the labels.</p>
EXIT STATUS	<p>The following exit values are returned:</p> <p>0 Write or display of labels was successful.</p> <p>non-zero An error occurred.</p>
FILES	<p>/etc/vfstab list of default parameters for each file system</p> <p>/etc/default/fs default local file system type. Default values can be set for the following flags in /etc/default/fs. For example: LOCAL=ufs</p> <p>LOCAL: The default partition for a command if no <i>FSType</i> is specified.</p>
SEE ALSO	volcopy (1M), vfstab (4) Manual pages for the <i>FSType</i> -specific modules of labelit

NOTES

This command may not be supported for all *FSTypes*.

NAME	labelit_hfs – provide and print labels for hfs file systems
SYNOPSIS	<code>/usr/sbin/labelit -F hfs [<i>generic_options</i>] [-o <i>specific_options</i>] <i>special</i></code>
DESCRIPTION	<p>labelit can be used to provide labels for unmounted CD-ROM images (CD-ROMs may not be labeled, as they are read-only media).</p> <p><i>generic_options</i> are options supported by the generic labelit command.</p> <p>If no <i>specific_options</i> are specified, labelit prints the current value of all label fields.</p> <p>The <i>special</i> name should be the physical disk section (for example, <code>/dev/dsk/c0d0s6</code>).</p>
OPTIONS	<p>-o Use one or more of the following <i>name=value</i> pairs separated by commas (with no intervening spaces) to specify values for specific label fields. According to the ISO 9660 specification, only certain sets of characters may be used to fill in these labels. Thus, “d-characters” below refers to the characters ‘A’ through ‘Z’, the digits ‘0’ through ‘9’, and the ‘_’ (underscore) character. “a-characters” below refers to ‘A’ through ‘Z’, ‘0’ through ‘9’, space, and the following characters: <code>!"%&'()*+,-./:;<=>?_.</code></p> <p>absfile= Abstract file identifier, d-characters, 37 characters maximum.</p> <p>applid= Application identifier, d-characters, 128 characters maximum.</p> <p>bibfile= Bibliographic file identifier, d-characters, 37 characters maximum.</p> <p>copyfile= Copyright file identifier, d-characters, 128 maximum.</p> <p>prepid= Data preparer identifier, d-characters, 128 maximum.</p> <p>pubid= Publisher identifier, d-characters, 128 maximum.</p> <p>sysid= System identifier, a-characters, 32 maximum.</p> <p>valid= Volume identifier, d-characters, 32 maximum.</p> <p>volsetid= Volume set identifier, d-characters, 128 maximum.</p>
SEE ALSO	labelit(1M) , volcopy(1M)

NAME	labelit_ufs – provide and print labels for ufs file systems
SYNOPSIS	labelit -F ufs [<i>generic_options</i>] <i>special</i> [<i>fsname volume</i>]
DESCRIPTION	labelit is used to write labels on unmounted disk file systems. Such labels may be used to uniquely identify volumes and are used by volume-oriented programs such as volcopy (1M).
OPTIONS	The following option is supported: <i>generic_options</i> options supported by the generic labelit command. See labelit (1M).
OPERANDS	The following operands are supported: <i>special</i> name should be the physical disk section (for example, /dev/dsk/c0d0s6). The device may not be on a remote machine. <i>fsname</i> represents the mount point (for example, root , u1 , and so on) of the file system. <i>volume</i> may be used to represent the physical volume name. If <i>fsname</i> and <i>volume</i> are not specified, labelit prints the current values of these labels. Both <i>fsname</i> and <i>volume</i> are limited to six or fewer characters.
EXIT STATUS	The following exit values are returned: 0 Write or display of labels was successful. non-zero An error occurred.
SEE ALSO	labelit (1M), volcopy (1M), fs_ufs (4)

NAME	leoconfig – initialize the Leo (ZX and TZX) Graphics Accelerator and download microcode
SYNOPSIS	<pre> /usr/sbin/leoconfig [-c] [-h] [-i] [-v] [-d <i>device_filename</i>] [-f <i>filename</i>] [-g <i>gamma_value</i>] [-I <i>microcode_directory</i>] [-l <i>leofloat_enable</i>] [[-degamma8 on off] [-DEGAMMA8 on off]] [-m <i>monitor_type</i>] /usr/sbin/leoconfig -G <i>gamma_value</i> [-v] /usr/sbin/leoconfig -M <i>monitor_type</i> [-v] </pre>
AVAILABILITY	SUNWleow
DESCRIPTION	<p>The leoconfig command initializes the Leo (both ZX and TZX) Graphics Accelerator and downloads microcode from the host. leoconfig is normally run as a part of the <code>/etc/init.d/leoconfig</code> script to download the Leo microcode file and to complete Leo initialization. The standard version of the file <code>/etc/init.d/leoconfig</code> runs leoconfig with the following command line options:</p> <pre> /usr/sbin/leoconfig \${MONTYPE} -d \$devpath/\$inst -I /usr/lib </pre> <p>If another configuration is desired, edit the <code>leoconfig</code> script in the <code>/etc/init.d/leoconfig</code> file. It is inadvisable to run the leoconfig command while the configured device is being used; unpredictable results may occur. To run leoconfig, first bring the window system down. Only root can run this command.</p> <p>File names may be either relative or absolute path names. Relative path names are prepended with the path specified by <code>-I</code>, or the default path <code>/usr/lib</code>.</p>
OPTIONS	<p>The following options are supported:</p> <ul style="list-style-type: none"> <code>-c</code> Checks that the microcode has loaded correctly. <code>-d <i>device_filename</i></code> Specifies the Leo special file. The default is <code>/dev/fbs/leo0</code>. <code>-degamma8 on off</code> Specifies automatic inverse gamma correction of 8-bit indexed color maps. This allows color maps with built-in gamma correction to work properly on Leo. The default is on. <code>-DEGAMMA8 on off</code> Like the <code>-degamma8</code> option but only specifies automatic inverse gamma correction; does not otherwise initialize Leo or download microcode. <code>-f <i>filename</i></code> Specifies the microcode file. The default is leo.ucode. <code>-g <i>gamma_value</i></code> Specifies the gamma correction value. The default is 2.22. <code>-G <i>gamma_value</i></code> Loads the gamma correction table only; does not initialize Leo or download microcode. Any other option besides <code>-v</code> will cause an invalid option message

to be displayed.

- h** Display help menu of valid command arguments.
- i** Initialize the Leo system.
- I *microcode_directory***
Specifies the directory containing the microcode files. The default is **/usr/lib**.
- l *leofloat_enable***
Specifies the LeoFloat enable mask in decimal form (one bit per float). The default for model ZX is **15**. The default for model TZX is **255**.
- m *monitor_type***
Specifies the monitor type, where *monitor_type* must be one of the following:

1280_76	1280 × 1024 @ 76 Hz, non-interlaced.
1280_67	1280 × 1024 @ 67 Hz, non-interlaced.
1152_76	1152 × 900 @ 76 Hz, non-interlaced.
1152_66	1152 × 900 @ 66 Hz, non-interlaced.
1024_76	1024 × 768 @ 76 Hz, non-interlaced.
1024_60	1024 × 768 @ 60 Hz, non-interlaced.
stereo_108	960 × 680 @ 108 Hz, non-interlaced, stereo.
stereo_114	960 × 680 @ 112 Hz, non-interlaced, stereo.
pal	770 × 575 @ 50 Hz, interlaced (PAL).
ntsc	640 × 480 @ 60 Hz, interlaced (NTSC).
- M *monitor_type***
Specifies the monitor type only; does not initialize the Leo or download microcode. Any other option besides **-v** will cause an invalid option message to be displayed. The *monitor_type* value is the same as for the **-m** option.
- v** Verbose. Display detailed progress reports.

EXAMPLES

The following example switches the monitor type to the maximum resolution of 1280 × 1024 at 76 Hz:

```
example% /usr/sbin/leoconfig -M 1280_76 -v
```

EXIT STATUS

- 0** success
- 1** an error has occurred

FILES

/dev/fbs/leo0	device special file
leo.unicode	Leo microcode file
/usr/lib	directory that normally contains microcode file
/etc/init.d/leoconfig	local Leo initialization script

SEE ALSO

mmap(2), **fbio(7I)**, **leo(7D)**

NAME	link, unlink – link and unlink files and directories
SYNOPSIS	<i>/usr/sbin/link file1 file2</i> <i>/usr/sbin/unlink file</i>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>The link command is used to create a file name that points to another file. <i>file1</i> is the existing file, <i>file2</i> is the newly-created file. Linked files and directories can be removed by the unlink command; however, it is strongly recommended that the rm and rmdir commands be used instead of the unlink command.</p> <p>The only difference between ln and link and unlink is that the latter do exactly what they are told to do, abandoning all error checking. This is because they directly invoke the link and unlink system calls.</p>
SEE ALSO	ln(1) , rm(1) , link(2) , unlink(2)
NOTES	These commands can be run only by the super-user.

NAME	list_devices – list allocatable devices
SYNOPSIS	list_devices [-s] [-U uid] -I [device] list_devices [-s] [-U uid] -n [device] list_devices [-s] [-U uid] -u [device]
AVAILABILITY	The functionality described in this man page is available only if the Basic Security Module (BSM) has been enabled. See bsmconv(1M) for more information.
DESCRIPTION	<p>list_devices lists the allocatable devices in the system according to specified qualifications.</p> <p>The <i>device</i> and all device special files associated with the device are listed. The device argument is optional and if it is not present, all relevant devices are listed.</p>
OPTIONS	<p>-I [device] List the pathname(s) of the device special files associated with the device that are allocatable to the current process. If <i>device</i> is given, list only the files associated with the specified device.</p> <p>-n [device] List the pathname(s) of device special files associated with the device that are allocatable to the current process but are not currently allocated. If <i>device</i> is given, list only the files associated with that device.</p> <p>-s Silent. Suppresses any diagnostic output.</p> <p>-u [device] List the pathname(s) of device special files, associated with the device that are allocated to the owner of the current process. If <i>device</i> is given, list only the files associated with that device.</p> <p>-U uid Use the user ID <i>uid</i> instead of the real user ID of the current process when performing the list_devices operation. Only the super user can use this option.</p>
DIAGNOSTICS	list_devices returns a nonzero exit status in the event of an error.
FILES	/etc/security/device_allocate /etc/security/device_maps /etc/security/dev/* /usr/security/lib/*
SEE ALSO	allocate(1M) , bsmconv(1M) , deallocate(1M) , device_allocate(4) , device_maps(4)

NAME	listdgrp – lists members of a device group
SYNOPSIS	<code>/usr/bin/listdgrp dgroup ...</code>
DESCRIPTION	listdgrp displays the members of the device groups specified by the <i>dgroup</i> list.
ERRORS	This command will exit with one of the following values: 0 = successful completion of the task. 1 = command syntax incorrect, invalid option used, or internal error occurred. 2 = device group table could not be opened for reading. 3 = device group <i>dgroup</i> could not be found in the device group table.
EXAMPLES	To list the devices that belong to group partitions : <pre>example% listdgrp partitions root swap usr</pre>
FILES	<code>/etc/dgroup.tab</code>
SEE ALSO	<code>putdgrp(1M)</code>

NAME	listen – network listener daemon
SYNOPSIS	<code>/usr/lib/saf/listen [-m <i>devstem</i>] <i>net_spec</i></code>
DESCRIPTION	<p>The listen process “listens” to a network for service requests, accepts requests when they arrive, and invokes servers in response to those service requests. The network listener process may be used with any connection-oriented network (more precisely, with any connection-oriented transport provider) that conforms to the Transport Layer Interface (TLI) Specification.</p> <p>The listener internally generates a pathname for the minor device for each connection; it is this pathname that is used in the utmp entry for a service, if one is created. By default, this pathname is the concatenation of the prefix <code>/dev/netspec</code> with the decimal representation of the minor device number. In either case, the representation of the minor device number will be at least two digits (for example, 05 or 27), or longer when it is necessary to accommodate minor device numbers larger than 99.</p>
SERVER INVOCATION	<p>When a connection indication is received, the listener creates a new transport endpoint and accepts the connection on that endpoint. Before giving the file descriptor for this new connection to the server, any designated STREAMS modules are pushed and the configuration script is executed, (if one exists). This file descriptor is appropriate for use with either TLI (see t_sync(3N)) or the sockets interface library.</p> <p>By default, a new instance of the server is invoked for each connection. When the server is invoked, file descriptor 0 refers to the transport endpoint, and is open for reading and writing. File descriptors 1 and 2 are copies of file descriptor 0; no other file descriptors are open. The service is invoked with the user and group IDs of the user name under which the service was registered with the listener, and with the current directory set to the HOME directory of that user.</p> <p>Alternatively, a service may be registered so that the listener will pass connections to a standing server process through a FIFO or a named STREAM, instead of invoking the server anew for each connection. In this case, the connection is passed in the form of a file descriptor that refers to the new transport endpoint. Before the file descriptor is sent to the server, the listener interprets any configuration script registered for that service using doconfig(3N), although doconfig is invoked with both the NORUN and NOASSIGN flags. The server receives the file descriptor for the connection in a strecvfd structure using an L_RECVFD ioctl(2).</p> <p>For more details about the listener and its administration, see nlsadmin(1M).</p>
OPTIONS	<code>-mdevstem</code> The listener will use <i>devstem</i> as the prefix for the pathname.
FILES	<code>/etc/saf/pmtag/*</code>

SEE ALSO **nlsadmin(1M)**, **pmadm(1M)**, **sac(1M)**, **sacadm(1M)**, **ioctl(2)**, **doconfig(3N)**, **nlsgetcall(3N)**, **nlsprovider(3N)**, **t_sync(3N)**, **streamio(7I)**

System Administration Guide, Volume II

NOTES When passing a connection to a standing server, the user and group IDs contained in the **strecvfd** structure will be those for the listener (that is, they will both be 0); the user name under which the service was registered with the listener is not reflected in these IDs.

When operating multiple instances of the listener on a single transport provider, there is a potential race condition in the binding of addresses during initialization of the listeners, if any of their services have dynamically assigned addresses. This condition would appear as an inability of the listener to bind a static-address service to its otherwise valid address, and would result from a dynamic-address service having been bound to that address by a different instance of the listener.

NAME	lockd – network lock daemon
SYNOPSIS	<code>/usr/lib/nfs/lockd [-g <i>graceperiod</i>] [-t <i>timeout</i>]</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>lockd is part of the NFS lock manager, which supports record locking operations on NFS files. See fcntl(2) and lockf(3C). The lock manager provides two functions:</p> <ul style="list-style-type: none">• it forwards fcntl(2) locking requests for NFS mounted file systems to the lock manager on the NFS server• it generates local file locking operations in response to requests forwarded from lock managers running on NFS client machines. <p>State information kept by the lock manager about these locking requests can be lost if the lockd is killed or the operating system is rebooted. Some of this information can be recovered as follows. When the server lock manager restarts, it waits for a grace period for all client-site lock managers to submit reclaim requests. Client-site lock managers, on the other hand are notified by the status monitor daemon, statd(1M), of the restart and promptly resubmit previously granted lock requests.</p>
OPTIONS	<p><code>-g <i>graceperiod</i></code> <i>graceperiod</i> is the number of seconds that clients have to reclaim locks after the server reboots. The default is 45 seconds.</p> <p><code>-t <i>timeout</i></code> <i>timeout</i> is the number of seconds to wait before retransmitting a lock request to the remote server. The default value is 15 seconds.</p>
SEE ALSO	statd(1M) , fcntl(2) , lockf(3C)

NAME	lockfs – change or report file system locks
SYNOPSIS	<code>/usr/sbin/lockfs [-dehnuw] [-c <i>string</i>] [-a] [<i>file-system</i> ...]</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>lockfs is used to change and report the status of file system locks. lockfs reports the lock status and unlocks the file systems that were improperly left locked by an application such as ufsdump(1M). This could occur if ufsdump(1M) is killed using kill(1).</p> <p>Using lockfs to lock a file system is discouraged because this requires extensive knowledge of SunOS internals to be used effectively and correctly.</p> <p><i>file-system</i> is a list of pathnames separated by white space. If <i>file-system</i> is not specified, and -a is specified, lockfs is run on all mounted, UFS type file systems.</p> <p>When invoked with no arguments, lockfs lists the UFS file systems that are locked.</p>
OPTIONS	<p>You must be super-user to use any of the following options, with the exception of the -a option.</p> <ul style="list-style-type: none"> -a Apply command to all mounted, UFS type file systems. <i>file-system</i> is ignored when -a is specified. -d delete-lock (dlock) the specified <i>file-system</i>. dlock suspends access that could remove directory entries. -e error-lock (elock) the specified <i>file-system</i>. elock blocks all local access to the locked file system and returns EWOULDLOCK on all remote access. File systems are elocked by UFS on detection of internal inconsistency. They may only be unlocked after successful repair by fsck, which is usually done automatically (see mount_ufs(1M).) elocked file systems can be unmounted. -h Hard-lock (hlock) the specified <i>file-system</i>. hlock returns an error on every access to the locked file system, and cannot be unlocked. hlocked file systems can be unmounted. -n Name-lock (nlock) the specified <i>file-system</i>. nlock suspends accesses that could change or remove existing directories entries. -u Unlock (unlock) the specified <i>file-system</i>. unlock awakens suspended accesses. -w Write-lock (wlock) the specified <i>file-system</i>. wlock suspends writes that would modify the file system. Access times are not kept while a file system is write-locked. -c <i>string</i> Accept a string that is passed as the comment field. The -c only takes affect when the lock is being set using either the d, h, n, u, or w options.
EXAMPLES	<p>In the following examples, <i>filesystem</i> is the pathname of the mounted-on directory (mount point). Locktype is one of “<i>write</i>,” “<i>name</i>,” “<i>delete</i>,” “<i>hard</i>,” or “<i>unlock</i>.” When enclosed in parenthesis, the lock is being set. Comment is a string set by the process that last issued a lock command.</p>

The following example shows the **lockfs** output when only the **-a** option is specified.

```
example# /usr/sbin/lockfs -a
Filesystem Locktype Comment
/          unlock
/var       unlock
example#
```

The following example shows the **lockfs** output when the **-w** option is used to write lock the **/var** file system and the comment string is set using the **-c** option. The **-a** option is then specified on a separate command line.

```
example# /usr/sbin/lockfs -w -c "lockfs: write lock example" /var
example# /usr/sbin/lockfs -a
Filesystem Locktype Comment
/          unlock
/var       write      lockfs: write lock example
example#
```

The following example shows the **lockfs** output when the **-u** option is used to unlock the **/var** file system and the comment string is set using the **-c** option.

```
example# /usr/sbin/lockfs -uc "lockfs: unlock example" /var
example# /usr/sbin/lockfs /var
Filesystem Locktype Comment
/var       unlock    lockfs: unlock example
example#
```

SEE ALSO

kill(1), **mount_ufs(1M)**, **ufsdump(1M)**, **fs_ufs(4)**

System Administration Guide, Volume 1

DIAGNOSTICS

file system: Not owner

You must be root to use this command.

file system :Deadlock condition detected/avoided

A file is enabled for accounting or swapping, on *file system*.

file system: Device busy

Another process is setting the lock on *file system*.

NAME	logins – list user and system login information
SYNOPSIS	<code>/usr/bin/logins [-admopstux] [-g group...] [-l login...]</code>
DESCRIPTION	This command displays information on user and system logins known to the system. Contents of the output is controlled by the command options and can include the following: user or system login, user id number, passwd account field value (user name or other information), primary group name, primary group id, multiple group names, multiple group ids, home directory, login shell, and four password aging parameters. The default information is the following: login id, user id, primary group name, primary group id and the account field value. Output is sorted by user id, system logins, followed by user logins.
OPTIONS	<ul style="list-style-type: none"> -a Add two password expiration fields to the display. The fields show how many days a password can remain unused before it automatically becomes inactive, and the date that the password will expire. -d Selects logins with duplicate uids. -m Displays multiple group membership information. -o Formats output into one line of colon-separated fields. -p Selects logins with no passwords. -s Selects all system logins. -t Sorts output by login instead of by uid. -u Selects all user logins. -x Prints an extended set of information about each selected user. The extended information includes home directory, login shell and password aging information, each displayed on a separate line. The password information consists of password status (PS for password, NP for no password or LK for locked). If the login is passworded, status is followed by the date the password was last changed, the number of days required between changes, and the number of days allowed before a change is required. The password aging information shows the time interval that the user will receive a password expiration warning message (when logging on) before the password expires. -g group Selects all users belonging to group, sorted by login. Multiple groups can be specified as a comma-separated list. -l login Selects the requested login. Multiple logins can be specified as a comma-separated list. Depending on the nameservice lookup types set in <code>/etc/nsswitch.conf</code>, the information can come from the <code>/etc/passwd</code> and <code>/etc/shadow</code> files and other nameservices.
NOTES	Options may be used together. If so, any login that matches any criteria will be displayed. When the -l and -g options are combined, a user will only be listed once, even if the user belongs to more than one of the selected groups.

NAME lpadmin – configure the LP print service

SYNOPSIS **lpadmin** **-p** *printer options*
lpadmin **-x** *dest*
lpadmin **-d** [*dest*]
lpadmin **-S** *print-wheel* **-A** *alert-type* [**-W** *minutes*] [**-Q** *requests*]
lpadmin **-M** **-f** *form-name* [**-a** [**-o** *filebreak*]] [**-t** *tray-number*]]

AVAILABILITY SUNWlpu

DESCRIPTION **lpadmin** configures the LP print service by defining printers and devices. It is used to add and change printers, to remove printers from service, to set or change the system default destination, to define alerts for printer faults, and to mount print wheels.

OPTIONS

Adding or Changing
a Printer

The first form of the **lpadmin** command (**lpadmin** **-p** *printer options*) is used to configure a new printer or to change the configuration of an existing printer. When creating a new printer, one of three options (**-v**, **-U**, or **-s**) must be supplied. In addition, only one of the following may be supplied: **-e**, **-i**, or **-m**; if none of these three options is supplied, the model standard is used. The **-h** and **-l** options are mutually exclusive. Printer and class names may be no longer than 14 characters and must consist entirely of the characters **A-Z**, **a-z**, **0-9**, dash (-) and underscore (_). If **-s** is specified, the following options are invalid: **-A**, **-e**, **-F**, **-h**, **-i**, **-l**, **-M**, **-m**, **-o**, **-U**, **-v**, and **-W**.

The following *printer options* may appear in any order.

-A *alert-type* [**-W** *minutes*]

The **-A** option is used to define an alert that informs the administrator when a printer fault is detected, and periodically thereafter, until the printer fault is cleared by the administrator. The *alert-types* are:

mail Send the alert message using mail (see **mail(1)**) to the administrator.

write Write the message to the terminal on which the administrator is logged in. If the administrator is logged in on several terminals, one is chosen arbitrarily.

quiet Do not send messages for the current condition. An administrator can use this option to temporarily stop receiving further messages about a known problem. Once the fault has been cleared and printing resumes, messages will again be sent when another fault occurs with the printer.

showfault

Attempt to execute a fault handler on each system that has a print job in the queue. The fault handler is **/etc/lp/alerts/printer**. It is invoked with three parameters: *printer_name*, *date*, *file_name*. The *file_name* is the name of a file containing the fault message.

none Do not send messages; any existing alert definition for the printer will be removed. No alert will be sent when the printer faults until a different

alert-type (except **quiet**) is used.

shell-command

Run the *shell-command* each time the alert needs to be sent. The shell command should expect the message in standard input. If there are blank spaces embedded in the command, enclose the command in quotes. Note that the **mail** and **write** values for this option are equivalent to the values **mail user-name** and **write user-name** respectively, where *user-name* is the current name for the administrator. This will be the login name of the person submitting this command unless he or she has used the **su** command to change to another user ID. If the **su** command has been used to change the user ID, then the *user-name* for the new ID is used.

list Display the type of the alert for the printer fault. No change is made to the alert.

The message sent appears as follows:

The printer *printer* has stopped printing for the reason given below. Fix the problem and bring the printer back on line. Printing has stopped, but will be restarted in a few minutes; issue an enable command if you want to restart sooner. Unless someone issues a change request

lp -i request-id -P ...

to change the page list to print, the current request will be reprinted from the beginning.

The reason(s) it stopped (multiple reasons indicate reprinted attempts):

reason

The LP print service can detect printer faults only through an adequate fast filter and only when the standard interface program or a suitable customized interface program is used. Furthermore, the level of recovery after a fault depends on the capabilities of the filter.

If the *printer* is **all**, the alerting defined in this command applies to all existing printers.

If the **-W** option is not used to arrange fault alerting for *printer*, the default procedure is to mail one message to the administrator of *printer* per fault. This is equivalent to specifying **-W once** or **-W 0**. If *minutes* is a number greater than zero, an alert will be sent at intervals specified by *minutes*.

-c class

Insert *printer* into the specified *class*. *class* will be created if it does not already exist.

-D comment

Save this *comment* for display whenever a user asks for a full description of *printer* (see **lpstat(1)**). The LP print service does not interpret this comment.

-e printer₁

Copy the interface program of an existing *printer₁* to be the interface program for

printer. (Options **-i** and **-m** may not be specified with this option.)

-F *fault-recovery*

This option specifies the recovery to be used for any print request that is stopped because of a printer fault, according to the value of *fault-recovery*:

continue Continue printing on the top of the page where printing stopped. This requires a filter to wait for the fault to clear before automatically continuing.

beginning Start printing the request again from the beginning.

wait Disable printing on *printer* and wait for the administrator or a user to enable printing again.

During the wait, the administrator or the user who submitted the stopped print request can issue a change request that specifies where printing should resume. (See the **-i** option of the **lp** command.) If no change request is made before printing is enabled, printing resumes at the top of the page where stopped, if the filter allows; otherwise, the request is printed from the beginning.

-f *allow:form-list*

-f *deny:form-list*

Allow or deny the forms in *form-list* to be printed on *printer*. By default no forms are allowed on a new printer.

For each printer, the LP print service keeps two lists of forms: an “allow-list” of forms that may be used with the printer, and a “deny-list” of forms that may not be used with the printer. With the **-f allow** option, the forms listed are added to the allow-list and removed from the deny-list. With the **-f deny** option, the forms listed are added to the deny-list and removed from the allow-list.

If the allow-list is not empty, only the forms in the list may be used on the printer, regardless of the contents of the deny-list. If the allow-list is empty, but the deny-list is not, the forms in the deny-list may not be used with the printer. All forms can be excluded from a printer by specifying **-f deny:all**. All forms can be used on a printer (provided the printer can handle all the characteristics of each form) by specifying **-f allow:all**.

The LP print service uses this information as a set of guidelines for determining where a form can be mounted. Administrators, however, are not restricted from mounting a form on any printer. If mounting a form on a particular printer is in disagreement with the information in the allow-list or deny-list, the administrator is warned but the mount is accepted. Nonetheless, if a user attempts to issue a print or change request for a form and printer combination that is in disagreement with the information, the request is accepted only if the form is currently mounted on the printer. If the form is later unmounted before the request can print, the request is canceled and the user is notified by mail.

If the administrator tries to specify a form as acceptable for use on a printer that doesn't have the capabilities needed by the form, the command is rejected.

Note the other use of **-f**, with the **-M** option, below.

The **-T** option must be invoked first with **lpadmin** to identify the printer type before the **-f** option can be used.

- h** Indicate that the device associated with the printer is hardwired. If neither of the mutually exclusive options, **-h** and **-I**, is specified, **-h** is assumed.

-I content-type-list

Allow *printer* to handle print requests with the content types listed in a *content-type-list*. If the list includes names of more than one type, the names must be separated by commas or blank spaces. (If they are separated by blank spaces, the entire list must be enclosed in double quotes.)

The type **simple** is recognized as the default content type for files in the UNIX system. A **simple** type of file is a data stream containing only printable ASCII characters and the following control characters.

Control Character	Octal Value	Meaning
backspace	10 ₈	move back one character, except at beginning of line
tab	11 ₈	move to next tab stop
linefeed (newline)	12 ₈	move to beginning of next line
form feed	14 ₈	move to beginning of next page
carriage return	15 ₈	move to beginning of current line

To prevent the print service from considering **simple** a valid type for the printer, specify either an explicit value (such as the printer type) in the *content-type-list*, or an empty list. If you do want **simple** included along with other types, you must include **simple** in the *content-type-list*.

Except for **simple**, each *content-type* name is freely determined by the administrator. If the printer type is specified by the **-T** option, then the printer type is implicitly considered to be also a valid content type.

-i interface

Establish a new interface program for *printer*. *interface* is the pathname of the new program. (The **-e** and **-m** options may not be specified with this option.)

- l** Indicate that the device associated with *printer* is a login terminal. The LP scheduler (**lpsched**) disables all login terminals automatically each time it is started. (The **-h** option may not be specified with this option.)

-M -f form-name [-a [-o filebreak]] [-t tray-number]]

Mount the form *form-name* on *printer*. Print requests that need the pre-printed form *form-name* will be printed on *printer*. If more than one printer has the form mounted and the user has specified **any** (with the **-d** option of the **lp** command) as the printer destination, then the print request will be printed on the one printer that also meets the other needs of the request.

The page length and width, and character and line pitches needed by the form are compared with those allowed for the printer, by checking the capabilities in the **terminfo** database for the type of printer. If the form requires attributes that are not available with the printer, the administrator is warned but the mount is accepted. If the form lists a print wheel as mandatory, but the print wheel mounted on the printer is different, the administrator is also warned but the mount is accepted.

If the **-a** option is given, an alignment pattern is printed, preceded by the same initialization of the physical printer that precedes a normal print request, with one exception: no banner page is printed. Printing is assumed to start at the top of the first page of the form. After the pattern is printed, the administrator can adjust the mounted form in the printer and press return for another alignment pattern (no initialization this time), and can continue printing as many alignment patterns as desired. The administrator can quit the printing of alignment patterns by typing **q**.

If the **-o filebreak** option is given, a formfeed is inserted between each copy of the alignment pattern. By default, the alignment pattern is assumed to correctly fill a form, so no formfeed is added.

If the **-t tray-number** option is specified, printer tray *tray-number* will be used.

A form is “unmounted” either by mounting a new form in its place or by using the **-f none** option. By default, a new printer has no form mounted.

Note the other use of **-f** without the **-M** option above.

-M -S print-wheel

Mount the *print-wheel* on *printer*. Print requests that need the *print-wheel* will be printed on *printer*. If more than one printer has *print-wheel* mounted and the user has specified **any** (with the **-d** option of the **lp** command) as the printer destination, then the print request will be printed on the one printer that also meets the other needs of the request.

If the *print-wheel* is not listed as acceptable for the printer, the administrator is warned but the mount is accepted. If the printer does not take print wheels, the command is rejected.

A print wheel is “unmounted” either by mounting a new print wheel in its place or by using the option **-S none**. By default, a new printer has no print wheel mounted.

Note the other uses of the **-S** option without the **-M** option described below.

-m model

Select *model* interface program, provided with the LP print service, for the printer. (Options **-e** and **-i** may not be specified with this option.)

-o option

Each **-o option** in the list below is the default given to an interface program if the option is not taken from a preprinted form description or is not explicitly given by the user submitting a request (see **lp(1)**). The only **-o options** that can have defaults defined are as follows:

```
length=scaled-decimal-number
width=scaled-decimal-number
cpi=scaled-decimal-number
lpi=scaled-decimal-number
stty= 'stty-option-list'
```

The term *scaled-decimal-number* refers to a non-negative number used to indicate a unit of size. The type of unit is shown by a “trailing” letter attached to the number. Three types of scaled decimal numbers can be used with the LP print service: numbers that show sizes in centimeters (marked with a trailing **c**); numbers that show sizes in inches (marked with a trailing **i**); and numbers that show sizes in units appropriate to use (without a trailing letter), that is, lines, characters, lines per inch, or characters per inch.

The first four default option values must agree with the capabilities of the type of physical printer, as defined in the **terminfo** database for the printer type. If they do not, the command is rejected.

The *stty-option-list* is not checked for allowed values, but is passed directly to the **stty** program by the standard interface program. Any error messages produced by **stty** when a request is processed (by the standard interface program) are mailed to the user submitting the request.

For each *option* not specified, the defaults for the following attributes are defined in the **terminfo** entry for the specified printer type.

```
length
width
cpi
lpi
```

The default for **stty** is

```
stty= '9600 cs8 -cstopb -parenb ixon
      -ixany opost -olcuc onlcr -ocrnl -onocr
      -onlret -ofill nl0 cr0 tab0 bs0 vt0 ff0'
```

You can set any of the **-o** options to the default values (which vary for different types of printers), by typing them without assigned values, as follows:

length=
width=
cpi=
lpi=
stty=

-o nobanner

Allow a user to submit a print request specifying that no banner page be printed.

-o banner

Force a banner page to be printed with every print request, even when a user asks for no banner page. This is the default; you must specify **-o nobanner** if you want to allow users to be able to specify **-o nobanner** with the **lp** command.

-P paper-name

Specify a paper type list that the printer supports.

-r class

Remove *printer* from the specified *class*. If *printer* is the last member of *class*, then *class* will be removed.

-S list

Allow either the print wheels or aliases for character sets named in *list* to be used on the printer.

If the printer is a type that takes print wheels, then *list* is a comma or space separated list of print wheel names. (Enclose the list with quotes if it contains blank spaces.) These will be the only print wheels considered mountable on the printer. (You can always force a different print wheel to be mounted.) Until the option is used to specify a list, no print wheels will be considered mountable on the printer, and print requests that ask for a particular print wheel with this printer will be rejected.

If the printer is a type that has selectable character sets, then *list* is a comma or blank separated list of character set name "mappings" or aliases. (Enclose the list with quotes if it contains blank spaces.) Each "mapping" is of the form

known-name=alias

The *known-name* is a character set number preceded by **cs** (such as **cs3** for character set three) or a character set name from the **terminfo** database entry **csnm**. See **terminfo(4)**. If this option is not used to specify a list, only the names already known from the **terminfo** database or numbers with a prefix of **cs** will be acceptable for the printer.

If *list* is the word **none**, any existing print wheel lists or character set aliases will be removed.

Note the other uses of the **-S** with the **-M** option described above.

The **-T** option must be invoked first with **lpadmin** to identify the printer type before the **-S** option can be used.

-s *system-name*!*printer-name*

Make a remote printer (one that must be accessed through another system) accessible to users on your system. *system-name* is the name of the remote system on which the remote printer is located; it must be listed in the systems table (**/etc/lp/Systems**). *printer-name* is the name used on the remote system for that printer. For example, if you want to access *printer₁* on *system₁* and you want it called *printer₂* on your system:

-p *printer₂* **-s** *system₁*!*printer₁*

-T *printer-type-list*

Identify the printer as being of one or more *printer-types*. Each *printer-type* is used to extract data from the **terminfo** database; this information is used to initialize the printer before printing each user's request. Some filters may also use a *printer-type* to convert content for the printer. If this option is not used, the default *printer-type* will be **unknown**; no information will be extracted from **terminfo** so each user request will be printed without first initializing the printer. Also, this option must be used if the following are to work: **-o cpi**, **-o lpi**, **-o width**, and **-o length** options of the **lpadmin** and **lp** commands, and the **-S** and **-f** options of the **lpadmin** command.

If the *printer-type-list* contains more than one type, then the *content-type-list* of the **-I** option must either be specified as **simple**, as empty (**-I ""**), or not specified at all.

-t *number-of-trays*

Specify the number of trays when creating the printer.

-u *allow:login-ID-list*

-u *deny:login-ID-list*

Allow or deny the users in *login-ID-list* access to the printer. By default all users are allowed on a new printer. The *login-ID-list* argument may include any or all of the following constructs:

<i>login-ID</i>	a user on any system
<i>system-name</i> ! <i>login-ID</i>	a user on system <i>system-name</i>
<i>system-name</i> ! all	all users on system <i>system-name</i>
all ! <i>login-ID</i>	a user on all systems
all	all users on all systems

For each printer, the LP print service keeps two lists of users: an "allow-list" of people allowed to use the printer, and a "deny-list" of people denied access to the printer. With the **-u allow** option, the users listed are added to the allow-list and removed from the deny-list. With the **-u deny** option, the users listed are added to the deny-list and removed from the allow-list.

If the allow-list is not empty, only the users in the list may use the printer, regardless of the contents of the deny-list. If the allow-list is empty, but the deny-list is

not, the users in the deny-list may not use the printer. All users can be denied access to the printer by specifying **-u deny:all**. All users may use the printer by specifying **-u allow:all**.

-U dial-info

The **-U** option allows your print service to access a remote printer. (It does not enable your print service to access a remote printer service.) Specifically, **-U** assigns the "dialing" information *dial-info* to the printer. *dial-info* is used with the **dial** routine to call the printer. Any network connection supported by the Basic Networking Utilities will work. *dial-info* can be either a phone number for a modem connection, or a system name for other kinds of connections. Or, if **-U direct** is given, no dialing will take place, because the name **direct** is reserved for a printer that is directly connected. If a system name is given, it is used to search for connection details from the file **/etc/uucp/Systems** or related files. The Basic Networking Utilities are required to support this option. By default, **-U direct** is assumed.

-v device

Associate a *device* with *printer*. *device* is the path name of a file that is writable by **lp**. Note that the same *device* can be associated with more than one printer.

**Removing a Printer
Destination**

The **-x dest** option removes the destination *dest* (a printer or a class), from the LP print service. If *dest* is a printer and is the only member of a class, then the class will be deleted, too. If *dest* is **all**, all printers and classes are removed. No other *options* are allowed with **-x**.

**Setting/Changing the
System Default
Destination**

The **-d [dest]** option makes *dest* (an existing printer or class) the new system default destination. If *dest* is not supplied, then there is no system default destination. No other *options* are allowed with **-d**.

**Setting an Alert for a
Print Wheel**

-S print-wheel -A alert-type [-W minutes] [-Q requests]

The **-S print-wheel** option is used with the **-A alert-type** option to define an alert to mount the print wheel when there are jobs queued for it. If this command is not used to arrange alerting for a print wheel, no alert will be sent for the print wheel. Note the other use of **-A**, with the **-p** option, above.

The *alert-types* are:

- mail** Send the alert message using the **mail** command to the administrator.
- write** Write the message, using the **write** command, to the terminal on which the administrator is logged in. If the administrator is logged in on several terminals, one is arbitrarily chosen.
- quiet** Do not send messages for the current condition. An administrator can use this option to temporarily stop receiving further messages about a known problem. Once the *print-wheel* has been mounted and subsequently unmounted, messages will again be sent when the number of print requests reaches the threshold specified by the **-Q** option.
- none** Do not send messages until the **-A** option is given again with a different *alert-type* (other than **quiet**).

shell-command

Run the *shell-command* each time the alert needs to be sent. The shell command should expect the message in standard input. If there are blanks embedded in the command, enclose the command in quotes. Note that the **mail** and **write** values for this option are equivalent to the values **mail *user-name*** and **write *user-name*** respectively, where *user-name* is the current name for the administrator. This will be the login name of the person submitting this command unless he or she has used the **su** command to change to another user ID. If the **su** command has been used to change the user ID, then the *user-name* for the new ID is used.

list Display the type of the alert for the print wheel on standard output. No change is made to the alert.

The message sent appears as follows:

**The print wheel *print-wheel* needs to be mounted
on the printer(s):**
printer (*integer*₁ requests)
*integer*₂ print requests await this print wheel.

The printers listed are those that the administrator had earlier specified were candidates for this print wheel. The number *integer*₁ listed next to each printer is the number of requests eligible for the printer. The number *integer*₂ shown after the printer list is the total number of requests awaiting the print wheel. It will be less than the sum of the other numbers if some requests can be handled by more than one printer.

If the *print-wheel* is **all**, the alerting defined in this command applies to all print wheels already defined to have an alert.

If the **-W** option is not given, the default procedure is that only one message will be sent per need to mount the print wheel. Not specifying the **-W** option is equivalent to specifying **-W once** or **-W 0**. If *minutes* is a number greater than zero, an alert will be sent at intervals specified by *minutes*.

If the **-Q** option is also given, the alert will be sent when a certain number (specified by the argument *requests*) of print requests that need the print wheel are waiting. If the **-Q** option is not given, or *requests* is 1 or **any** (which are both the default), a message is sent as soon as anyone submits a print request for the print wheel when it is not mounted.

FILES

/var/spool/lp/*
/etc/lp
/etc/lp/alerts/printer fault handler for **lpadmin**.

SEE ALSO

enable(1), **lp(1)**, **lpstat(1)**, **stty(1)**, **accept(1M)**, **lpsched(1M)**, **lpssystem(1M)**, **dial(3N)**, **terminfo(4)**

System Administration Guide, Volume II

NAME	lpfilter – administer filters used with the LP print service
SYNOPSIS	<code>/usr/sbin/lpfilter -f filter-name -F path-name</code> <code>/usr/sbin/lpfilter -f filter-name - -i -x -l</code>
AVAILABILITY	SUNWlpu
DESCRIPTION	<p>The lpfilter command is used to add, change, delete, and list a filter used with the LP print service. These filters convert the content type of a file to a content type acceptable to a printer.</p> <p>The argument all can be used instead of a <i>filter-name</i> with any of these options. When all is specified with the -F or - option, the requested change is made to all filters. Using all with the -i option has the effect of restoring to their original settings all filters for which predefined settings were initially available. Using the all argument with the -x option results in all filters being deleted, and using it with the -l option produces a list of all filters.</p>
OPTIONS	<p>-F path-name To add or change a filter. (-F path-name or - for standard input).</p> <p>-f filter-name Adds <i>filter-name</i> to the filter table.</p> <p>-i To reset an original filter to its factory setting.</p> <p>-x To delete a filter.</p> <p>-l To list a filter description.</p>
USAGE Adding or Changing a Filter	<p>The filter named in the -f option is added to the filter table. If the filter already exists, its description is changed to reflect the new information in the input.</p> <p>The filter description is taken from the <i>path-name</i> if the -F option is given, or from standard input if the - option is given. One of the two must be given to define or change a filter. If the filter named is one originally delivered with the LP print service, the -i option will restore the original filter description.</p> <p>When an existing filter is changed with the -F or - option, items that are not specified in the new information are left as they were. When a new filter is added with this command, unspecified items are given default values. (See below.)</p> <p>Filters are used to convert the content of a request into a data stream acceptable to a printer. For a given print request, the LP print service will know the following:</p> <ul style="list-style-type: none"> • the type of content in the request, • the name of the printer, • the type of the printer, • the types of content acceptable to the printer, and • the modes of printing asked for by the originator of the request.

It uses this information to find a filter or a pipeline of filters that will convert the content into a type acceptable to the printer.

Below is a list of items that provide input to this command, and a description of each item. All lists are comma or space separated.

Input types: *content-type-list*
Output types: *content-type-list*
Printer types: *printer-type-list*
Printers: *printer-list*
Filter type: *filter-type*
Command: *shell-command*
Options: *template-list*

Input types	This gives the types of content that can be accepted by the filter. (The default is any .)
Output types	This gives the types of content that the filter can produce from any of the input content types. (The default is any .)
Printer types	This gives the type of printers for which the filter can be used. The LP print service will restrict the use of the filter to these types of printers. (The default is any .)
Printers	This gives the names of the printers for which the filter can be used. The LP print service will restrict the use of the filter to just the printers named. (The default is any .)
Filter type	This marks the filter as a slow filter or a fast filter. Slow filters are generally those that take a long time to convert their input. They are run while unconnected to a printer, to keep the printers from being tied up while the filter is running. If a listed printer is on a remote system, the filter type for it must have the value slow . Fast filters are generally those that convert their input quickly, or those that must be connected to the printer when run. These will be given to the interface program to run while connected to the physical printer.
Command	This specifies which program to run to invoke the filter. The full program pathname as well as fixed options must be included in the <i>shell-command</i> ; additional options are constructed, based on the characteristics of each print request and on the Options field. A command must be given for each filter. The command must accept a data stream as standard input and produce the converted data stream on its standard output. This allows filter pipelines to be constructed to convert data not handled by a single filter.

Options

This is a comma separated list of templates used by the LP print service to construct options to the filter from the characteristics of each print request listed in the table later.

In general, each template is of the following form:

keyword pattern = replacement

The *keyword* names the characteristic that the template attempts to map into a filter-specific option; each valid *keyword* is listed in the table below.

A *pattern* is one of the following: a literal pattern of one of the forms listed in the table, a single asterisk (*), or a regular expression. If *pattern* matches the value of the characteristic, the template fits and is used to generate a filter specific option.

The *replacement* is what will be used as the option.

Regular expressions are the same as those found on the **regexp(5)** manual page. This includes the `\(...\)` and `\n` constructions, which can be used to extract portions of the *pattern* for copying into the *replacement*, and the `&`, which can be used to copy the entire *pattern* into the *replacement*.

The *replacement* can also contain a *; it too, is replaced with the entire *pattern*, just like the `&` of **regexp(5)**.

lp Option	Characteristic	keyword	Possible patterns
-T	Content type (input)	INPUT	<i>content-type</i>
N/A	Content type (output)	OUTPUT	<i>content-type</i>
N/A	Printer type	TERM	<i>printer-type</i>
-d	Printer name	PRINTER	<i>printer-name</i>
-f, -o cpi=	Character pitch	CPI	<i>integer</i>
-f, -o lpi=	Line pitch	LPI	<i>integer</i>
-f, -o length=	Page length	LENGTH	<i>integer</i>
-f, -o width=	Page width	WIDTH	<i>integer</i>
-P	Pages to print	PAGES	<i>page-list</i>
-S	Character set	CHARSET	<i>character-set-name</i>
	Print wheel	CHARSET	<i>print-wheel-name</i>
-f	Form name	FORM	<i>form-name</i>
-y	Modes	MODES	<i>mode</i>
-n	Number of copies	COPIES	<i>integer</i>

EXAMPLES

For example, the template

MODES landscape = -l

shows that if a print request is submitted with the **-y landscape** option, the filter will be given the option **-l**. As another example, the template

TERM * = -T *

shows that the filter will be given the option **-T printer-type** for whichever *printer-type* is associated with a print request using the filter.

As a last example, consider the template

MODES prwidth\=\(.*\)= -w\1

Suppose a user gives the command

lp -y prwidth=10

From the table above, the LP print service determines that the **-y** option is handled by a **MODES** template. The **MODES** template here works because the *pattern* **prwidth\=\(.*\)** matches the **prwidth=10** given by the user. The *replacement* **-w\1** causes the LP print service to generate the filter option **-w10**.

If necessary, the LP print service will construct a filter pipeline by concatenating several filters to handle the user's file and all the print options. (See **sh(1)** for a description of a pipeline.) If the print service constructs a filter pipeline, the **INPUT** and **OUTPUT** values used for each filter in the pipeline are the types of input and output for that filter, not for the entire pipeline.

Deleting a Filter

The **-x** option is used to delete the filter specified in *filter-name* from the LP filter table.

Listing a Filter Description

The **-l** option is used to list the description of the filter named in *filter-name*. If the command is successful, the following message is sent to standard output:

Input types: *content-type-list*
Output types: *content-type-list*
Printer types: *printer-type-list*
Printers: *printer-list*
Filter type: *filter-type*
Command: *shell-command*
Options: *template-list*

If the command fails, an error message is sent to standard error.

SEE ALSO

lp(1), **lpadmin(1M)**, **regexp(5)**

System Administration Guide, Volume II

NAME lpforms – administer forms used with the LP print service

SYNOPSIS **lpforms** *-f form-name option*
lpforms *-f form-name -A alert-type [-P paper-name [-d]] [-Q requests]*
[-W minutes]

AVAILABILITY SUNWlpu

DESCRIPTION The **lpforms** command administers the use of preprinted forms, such as company letter-head paper, with the LP print service. A form is specified by its *form-name*. Users may specify a form when submitting a print request (see **lp(1)**). The argument **all** can be used instead of *form-name* with either of the command lines shown above. The first command line allows the administrator to add, change, and delete forms, to list the attributes of an existing form, and to allow and deny users access to particular forms. The second command line is used to establish the method by which the administrator is alerted that the form *form-name* must be mounted on a printer.

OPTIONS *-f formname* Specify a form.

The first form of **lpforms** requires that one of the following *option* (*-*, *-l*, *-F*, *-x*) must be used:

- F pathname* To add or change form *form-name*, as specified by the information in *pathname*.
- To add or change form *form-name*, as specified by the information from standard input.
- x* To delete form *form-name* (this option must be used separately; it may not be used with any other option).
- l* To list the attributes of form *form-name*.

The second form of the **lpforms** command requires the *-A alert-type* option. The other options are optional.

- A alert-type* Defines an alert to mount the form when there are queued jobs which need it.
- P paper-name [-d]* Specify the paper name when creating the form. If *-d* is specified, this paper is the default.
- Q requests* An alert will be sent when a certain number of print requests that need the form are waiting.
- W minutes* An alert will be sent at intervals specified by minutes.

USAGE
**Adding or Changing
a Form**

The *-F pathname* option is used to add a new form, *form-name*, to the LP print service, or to change the attributes of an existing form. The form description is taken from *pathname* if the *-F* option is given, or from the standard input if the *-* option is used. One of these

two options must be used to define or change a form.

pathname is the path name of a file that contains all or any subset of the following information about the form.

Page length: *scaled-decimal-number*₁

Page width: *scaled-decimal-number*₂

Number of pages: *integer*

Line pitch: *scaled-decimal-number*₃

Character pitch: *scaled-decimal-number*₄

Character set choice: *character-set/print-wheel* [mandatory]

Ribbon color: *ribbon-color*

Comment:

comment

Alignment pattern: [*content-type*]

content

The term “scaled-decimal-number” refers to a non-negative number used to indicate a unit of size. The type of unit is shown by a “trailing” letter attached to the number. Three types of scaled decimal numbers can be used with the LP print service: numbers that show sizes in centimeters (marked with a trailing *c*); numbers that show sizes in inches (marked with a trailing *i*); and numbers that show sizes in units appropriate to use (without a trailing letter); lines, characters, lines per inch, or characters per inch.

Except for the last two lines, the above lines may appear in any order. The **Comment:** and *comment* items must appear in consecutive order but may appear before the other items, and the **Alignment pattern:** and the *content* items must appear in consecutive order at the end of the file. Also, the *comment* item may not contain a line that begins with any of the key phrases above, unless the key phrase is preceded with a > sign. Any leading > sign found in the *comment* will be removed when the comment is displayed. There is no case distinction among the key phrases.

When this command is issued, the form specified by *form-name* is added to the list of forms. If the form already exists, its description is changed to reflect the new information. Once added, a form is available for use in a print request, except where access to the form has been restricted, as described under the **-u** option. A form may also be allowed to be used on certain printers only.

A description of each form attribute is below:

Page length and Page Width

Before printing the content of a print request needing this form, the generic interface program provided with the LP print service will initialize the physical printer to handle pages *scaled-decimal-number*₁ long, and *scaled-decimal-number*₂ wide using the printer type as a key into the **terminfo(4)** database.

The page length and page width will also be passed, if possible, to each filter used in a request needing this form.

Number of pages

Each time the alignment pattern is printed, the LP print service will attempt to truncate the *content* to a single form by, if possible, passing to each filter the page subset of 1-*integer*.

Line pitch and Character pitch

Before printing the content of a print request needing this form, the interface program provided with the LP print service will initialize the physical printer to handle these pitches, using the printer type as a key into the **terminfo**(4) database. Also, the pitches will be passed, if possible, to each filter used in a request needing this form. *scaled-decimal-number*₃ is in lines-per-centimeter if a *c* is appended, and lines-per-inch otherwise; similarly, *scaled-decimal-number*₄ is in characters-per-centimeter if a *c* is appended, and characters-per-inch otherwise. The character pitch can also be given as **elite** (12 characters-per-inch), **pica** (10 characters-per-inch), or **compressed** (as many characters-per-inch as possible).

Character set choice

When the LP print service alerts an administrator to mount this form, it will also mention that the print wheel *print-wheel* should be used on those printers that take print wheels. If printing with this form is to be done on a printer that has selectable or loadable character sets instead of print wheels, the interface programs provided with the LP print service will automatically select or load the correct character set. If **mandatory** is appended, a user is not allowed to select a different character set for use with the form; otherwise, the character set or print wheel named is a suggestion and a default only.

Ribbon color

When the LP print service alerts an administrator to mount this form, it will also mention that the color of the ribbon should be *ribbon-color*.

Comment

The LP print service will display the *comment* unaltered when a user asks about this form (see **lpstat**(1)).

Alignment pattern

When mounting this form, an administrator can ask for the *content* to be printed repeatedly, as an aid in correctly positioning the preprinted form. The optional *content-type* defines the type of printer for which *content* had been generated. If *content-type* is not given, **simple** is assumed. Note that the *content* is stored as given, and will be readable only by the user **lp**.

When an existing form is changed with this command, items missing in the new information are left as they were. When a new form is added with this command, missing items will get the following defaults:

Page Length: **66**
 Page Width: **80**
 Number of Pages: **1**
 Line Pitch: **6**
 Character Pitch: **10**
 Character Set Choice: **any**
 Ribbon Color: **any**

Deleting a Form

The **-x** option is used to delete the form *form-name* from the LP print service.

Listing Form Attributes

The **-l** option is used to list the attributes of the existing form *form-name*. The attributes listed are those described under **Adding and Changing a Form**, above. Because of the potentially sensitive nature of the alignment pattern, only the administrator can examine the form with this command. Other people may use the **lpstat(1)** command to examine the non-sensitive part of the form description.

Allowing and Denying Access to a Form

The **-u** option, followed by the argument **allow:login-ID-list** or **-u deny:login-ID-list** lets you determine which users will be allowed to specify a particular form with a print request. This option can be used with the **-F** or **-** option, each of which is described above under **Adding or Changing a Form**.

The *login-ID-list* argument may include any or all of the following constructs:

login-ID A user on any system
system_name!login-ID
 A user on system *system_name*
system_name!all All users on system *system_name*
all!login-ID A user on all systems
all All users on all systems

The LP print service keeps two lists of users for each form: an “allow-list” of people allowed to use the form, and a “deny-list” of people that may not use the form. With the **-u allow** option, the users listed are added to the allow-list and removed from the deny-list. With the **-u deny** option, the users listed are added to the deny-list and removed from the allow-list. (Both forms of the **-u** option can be run together with the **-F** or the **-** option.)

If the allow-list is not empty, only the users in the list are allowed access to the form, regardless of the content of the deny-list. If the allow-list is empty but the deny-list is not, the users in the deny-list may not use the form, (but all others may use it). All users can be denied access to a form by specifying **-f deny:all**. All users can be allowed access to a form by specifying **-f allow:all**. (This is the default.)

Setting an Alert to Mount a Form

The `-f form-name` option is used with the `-A alert-type` option to define an alert to mount the form when there are queued jobs which need it. If this option is not used to arrange alerting for a form, no alert will be sent for that form.

The method by which the alert is sent depends on the value of the `alert-type` argument specified with the `-A` option. The `alert-types` are:

- mail** Send the alert message using the **mail** command to the administrator.
- write** Write the message, using the **write** command, to the terminal on which the administrator is logged in. If the administrator is logged in on several terminals, one is arbitrarily chosen.
- quiet** Do not send messages for the current condition. An administrator can use this option to temporarily stop receiving further messages about a known problem. Once the form `form-name` has been mounted and subsequently unmounted, messages will again be sent when the number of print requests reaches the threshold specified by the `-Q` option.

showfault

Attempt to execute a form alert handler on each system that has a print job for that form in the queue. The fault handler is `/etc/lp/alerts/form`. It is invoked with three parameters: `form_name`, `date`, `file_name`. `file_name` is the name of a file containing the form alert message.

- none** Do not send messages until the `-A` option is given again with a different `alert-type` (other than **quiet**).

shell-command

Run the `shell-command` each time the alert needs to be sent. The shell command should expect the message in standard input. If there are blank spaces embedded in the command, enclose the command in quotes. Note that the **mail** and **write** values for this option are equivalent to the values **mail login-ID** and **write login-ID** respectively, where `login-ID` is the current name for the administrator. This will be the login name of the person submitting this command unless he or she has used the **su** command to change to another login-ID. If the **su** command has been used to change the user ID, then the `user-name` for the new ID is used.

- list** Display the type of the alert for the form on standard output. No change is made to the alert.

The message sent appears as follows:

The form `form-name` needs to be mounted on the printer(s):
printer (*integer*₁ requests).
*integer*₂ print requests await this form.
 Use the *ribbon-color* ribbon.
 Use the *print-wheel* print wheel, if appropriate.

The printers listed are those that the administrator has specified as candidates for this form. The number *integer₁* listed next to each printer is the number of requests eligible for the printer. The number *integer₂* shown after the list of printers is the total number of requests awaiting the form. It will be less than the sum of the other numbers if some requests can be handled by more than one printer. The *ribbon-color* and *print-wheel* are those specified in the form description. The last line in the message is always sent, even if none of the printers listed use print wheels, because the administrator may choose to mount the form on a printer that does use a print wheel.

Where any color ribbon or any print wheel can be used, the statements above will read:

Use any ribbon.

Use any print-wheel.

If *form-name* is **any**, the *alert-type* defined in this command applies to any form for which an alert has not yet been defined. If *form-name* is **all**, the *alert-type* defined in this command applies to all forms.

If the **-W** *minutes* option is not given, the default procedure is that only one message will be sent per need to mount the form. Not specifying the **-W** option is equivalent to specifying **-W once** or **-W 0**. If *minutes* is a number greater than **0**, an alert will be sent at intervals specified by *minutes*.

If the **-Q** *requests* option is also given, the alert will be sent when a certain number (specified by the argument *requests*) of print requests that need the form are waiting. If the **-Q** option is not given, or the value of *requests* is **1** or **any** (which are both the default), a message is sent as soon as anyone submits a print request for the form when it is not mounted.

Listing the Current Alert

The **-f** option, followed by the **-A** option and the argument **list** is used to list the *alert-type* that has been defined for the specified form *form-name*. No change is made to the alert. If *form-name* is recognized by the LP print service, one of the following lines is sent to the standard output, depending on the type of alert for the form.

- **When requests requests are queued:**
alert with shell-command every minutes minutes
- **When requests requests are queued:**
write to user-name every minutes minutes
- **When requests requests are queued:**
mail to user-name every minutes minutes
- **No alert**

The phrase **every minutes minutes** is replaced with **once** if *minutes* (**-W minutes**) is **0**.

Terminating an Active Alert

The **-A quiet** option is used to stop messages for the current condition. An administrator can use this option to temporarily stop receiving further messages about a known problem. Once the form has been mounted and then unmounted, messages will again be sent when the number of print requests reaches the threshold *requests*.

**Removing an Alert
Definition**

No messages will be sent after the **-A none** option is used until the **-A** option is given again with a different *alert-type*. This can be used to permanently stop further messages from being sent as any existing alert definition for the form will be removed.

FILES

/etc/lp/alerts/form fault handler for **lpform**.

SEE ALSO

lp(1), lpadmin(1M), lpstat(1), terminfo(4)
System Administration Guide, Volume II

NAME	lpsched , lpshut , lpmove – start/stop the LP print service and move requests
SYNOPSIS	/usr/lib/lp/lpsched lpshut lpmove requests dest lpmove dest₁ dest₂
DESCRIPTION	<p>lpsched starts the LP print service; this can be done only by root.</p> <p>lpshut shuts down the print service. All printers that are printing at the time lpshut is invoked will stop printing. When lpsched is started again, requests that were printing at the time a printer was shut down will be reprinted from the beginning.</p> <p>lpmove moves requests that were queued by lp(1) between LP destinations. lpmove can only be used among local printers.</p> <p>The first form of the lpmove command moves the named <i>requests</i> to the LP destination <i>dest</i>. <i>requests</i> are request-IDs as returned by lp. If a request was originally queued for a class, or the special destination any, the destination of the request will be changed to <i>dest</i>. The request will be printable only on <i>dest</i> and not on other members of the class or other acceptable printers.</p> <p>The second form of the lpmove command attempts to move all requests for destination <i>dest₁</i> to destination <i>dest₂</i>; lp will then reject any new requests for <i>dest₁</i>.</p> <p>Note that when moving requests, lpmove never checks the acceptance status of the new destination (see accept(1M)). Also, the request-IDs of the moved request are not changed, so that users can still find their requests. The lpmove command will not move requests that have options (content type, form required, and so on) that cannot be handled by the new destination.</p>
FILES	/var/spool/lp/*
SEE ALSO	enable(1) , lp(1) , lpstat(1) , accept(1M) , lpadmin(1M) <i>System Administration Guide, Volume II</i>

NAME	<code>lpsystem</code> – register remote systems with the print service
SYNOPSIS	<pre>lpsystem [-t <i>type</i>] [-T <i>timeout</i>] [-R <i>retry</i>] [-y "<i>comment</i>"] <i>system-name</i> [<i>system-name</i> ...] lpsystem -l [<i>system-name</i> ...] lpsystem -r <i>system-name</i> [<i>system-name</i> ...] lpsystem -A</pre>
AVAILABILITY	SUNWlpu
DESCRIPTION	<p>The lpsystem command is used to define parameters for the LP print service, with respect to communication with remote systems. Only a privileged user (that is, the owner of the login root) may execute the lpsystem command.</p> <p>Specifically, the lpsystem command is used to define remote systems with which the local LP print service can exchange print requests. These remote systems are described to the local LP print service in terms of several parameters that control communication: <i>type</i>, <i>retry</i> and <i>timeout</i>. These parameters are defined in <code>/etc/lp/Systems</code>. You can edit this file with a text editor (such as vi) but editing is not recommended.</p> <p>The <i>type</i> parameter defines the remote system as one of two types: s5 (SunOS 5.x operating system), or bsd. The default type is s5.</p> <p>The <i>timeout</i> parameter specifies the length of time (in minutes) that the print service should allow a network connection to be idle. If the connection to the remote system is idle (that is, there is no network traffic) for <i>N</i> minutes, then drop the connection. (When there is more work the connection will be re-established.) Legal values are n, 0, and <i>N</i>, where <i>N</i> is an integer greater than 0. The value n means “never time out”; 0 means “as soon as the connection is idle, drop it.” The default is n.</p> <p>The <i>retry</i> parameter specifies the length of time to wait before trying to re-establish a connection to the remote system, when the connection was dropped abnormally (that is, a network error). Legal values are n, 0, and <i>N</i>, where <i>N</i> is an integer greater than 0 and it means “wait <i>N</i> minutes before trying to reconnect. (The default is 10 minutes.) The value n means “do not retry dropped connections until there is more work”; 0 means “try to reconnect immediately.”</p> <p>The <i>comment</i> argument allows you to associate a free form comment with the system entry. This is visible when lpsystem -l is used.</p> <p><i>system-name</i> is the name of the remote system from which you want to be able to receive jobs, and to which you want to be able to send jobs. If the <i>system-name</i> is a plus sign (“+”), then anonymous client support is enabled. That is, your system will accept remote print jobs from any other print client (bsd or s5). This is enabled by default in <code>/etc/lp/Systems</code>; any other entries in the <code>/etc/lp/Systems</code> file will be superfluous. The other parameters listed on the line beginning with the <i>plus sign</i> are for reference only, and will not actually change the behavior of lpsched(1M).</p> <p>The command lpsystem -l [<i>system-name</i>] will print out a description of the parameters associated with <i>system-name</i> (if a system has been specified), or with all the systems in its database (if <i>system-name</i> has not been specified).</p>

The command **lpsystem -r *system-name*** will remove the entry associated with *system-name*. The print service will no longer accept jobs from that system or send jobs to it, even if the remote printer is still defined on the local system.

The command **lpsystem -A** will print out the TCP/IP address of the local machine in a format to be used when configuring the local port monitor to accept requests from a SunOS system.

OPTIONS

- t *type*** Specifies the remote system type.
- T *timeout*** Specifies the time allowed for a network connection to be idle. *timeout* is in minutes. Default is to never time out.
- R *retry*** Specifies time to wait before trying to reestablish a connection for a remote system.
- y *comment*** The comment argument allows you to associate a free form comment with the system entry.
- l [*system-name*]** Prints out a description of the parameters associated with *system-name*, or with all the systems in its database.
- r *system-name*** Removes the entry associated with *system-name*.
- A** Prints out the TCP/IP address in a format.

FILES

/var/spool/lp/ /etc/lp/**

SEE ALSO

lpsched(1M), **nlsadmin(1M)**, **sacadm(1M)**, **netdir(3N)**, **hosts(4)**, **netconfig(4)**, **services(4)**
NIS+ and FNS Administration Guide
System Administration Guide, Volume II

NOTES

With respect to */etc/lp/Systems*, this information is relatively minimal with respect to controlling network communications. Network addresses and services are handled by the **Netconfig** and **Netdir** facilities (see the *NIS+ and FNS Administration Guide* for a discussion of network addresses and services.) Port monitors handle listening for remote service requests and routing the connection to the print service (see the *Solaris 1.x to Solaris 2.x Transition Guide* for a discussion of port monitors.)

If the **Netconfig** and **Netdir** facilities are not set up properly, out-bound remote print service probably will not work. Similarly, if the local port monitors are not set up to route remote print requests to the print service, then service for remote systems will not be provided. See the chapters on managing printers in the *System Administration Guide, Volume II* for instructions.

With respect to the semantics of the *timeout* and *retry* values, the print service uses one process for each remote system with which it communicates, and it communicates with a remote system only when there is work to be done on that system or work being sent from that system.

The system initiating the connection is the “master” process and the system accepting the connection is the “slave” process. This designation serves only to determine which process dies (the slave) when a connection is dropped. This helps prevent there from being

more than one process communicating with a remote system. Furthermore, all connections are bi-directional, regardless of the master/slave designation. You cannot control a system's master/slave designation. Now, keeping all this information in mind, if a master process times out, then both the slave and master will exit. If a slave times out, then it is possible that the master may still live and retry the connection after the retry interval. Therefore, one system's resource management strategy can effect another system's strategy.

With respect to **lpsystem -A**: a SunOS 4.x system (described with **-t bsd**) can be connected to your system only via TCP/IP, and print requests from a SunOS system can come in to your machine only via a special port (515). The address given to you from **lpsystem** will be the address of your system and port 515. This address is used by your TCP/IP port monitor (see **sacadm(1M)** and **nlsadmin(1M)**) to "listen" on that address and port, and to route connections to the print service. (This procedure is discussed in the *System Administration Guide, Volume II*.) The important point here is that this is where you get the address referred to in that procedure.

The command **lpsystem -A** will not work if your system name and IP address are not listed in **/etc/inet/hosts**, (see **hosts(4)**), and the printer service is not listed in **/etc/inet/services**, (see **services(4)**).

The file **/etc/lp/Systems** is set by default to support anonymous print clients. This feature can be disabled if one wishes greater security for print jobs. However, it should be noted that this will increase the amount of work required of the system administrator. A good backup of this file is strongly recommended if anonymous print client support is disabled.

NAME	lpusers – set printing queue priorities										
SYNOPSIS	<p>lpusers -d <i>priority-level</i></p> <p>lpusers -q <i>priority-level</i> -u <i>login-ID-list</i></p> <p>lpusers -u <i>login-ID-list</i></p> <p>lpusers -q <i>priority-level</i></p> <p>lpusers -l</p>										
AVAILABILITY	SUNWlps										
DESCRIPTION	<p>The lpusers command sets limits to the queue priority level that can be assigned to jobs submitted by users of the LP print service.</p> <p>The first form of the command (with -d) sets the system-wide priority default to <i>priority-level</i>, where <i>priority-level</i> is a value of 0 to 39, with 0 being the highest priority. If a user does not specify a priority level with a print request (see lp(1)), the default priority level is used. Initially, the default priority level is 20.</p> <p>The second form of the command (with -q and -u) sets the default highest <i>priority-level</i> (0-39) that the users in <i>login-ID-list</i> can request when submitting a print request. The <i>login-ID-list</i> argument may include any or all of the following constructs:</p> <table border="0" style="margin-left: 40px;"> <tr> <td><i>login-ID</i></td> <td>A user on any system</td> </tr> <tr> <td><i>system_name!login-ID</i></td> <td>A user on the system <i>system_name</i></td> </tr> <tr> <td><i>system_name!all</i></td> <td>All users on system <i>system_name</i></td> </tr> <tr> <td>all!login-ID</td> <td>A user on all systems</td> </tr> <tr> <td>all</td> <td>All users on all systems</td> </tr> </table> <p>Users that have been given a limit cannot submit a print request with a higher priority level than the one assigned, nor can they change a request that has already been submitted to have a higher priority. Any print requests submitted with priority levels higher than allowed will be given the highest priority allowed.</p> <p>The third form of the command (with -u) removes any explicit priority level for the specified users.</p> <p>The fourth form of the command (with -q) sets the default highest priority level for all users not explicitly covered by the use of the second form of this command.</p> <p>The last form of the command (with -l) lists the default priority level and the priority limits assigned to users.</p>	<i>login-ID</i>	A user on any system	<i>system_name!login-ID</i>	A user on the system <i>system_name</i>	<i>system_name!all</i>	All users on system <i>system_name</i>	all!login-ID	A user on all systems	all	All users on all systems
<i>login-ID</i>	A user on any system										
<i>system_name!login-ID</i>	A user on the system <i>system_name</i>										
<i>system_name!all</i>	All users on system <i>system_name</i>										
all!login-ID	A user on all systems										
all	All users on all systems										
OPTIONS	<p>-d <i>priority-level</i> Set the system-wide priority default to <i>priority-level</i>.</p> <p>-q <i>priority-level</i> -u <i>login-ID-list</i> Set the default highest <i>priority-level</i> that the users in <i>login-ID-list</i> can request when submitting a print request.</p>										

- u** *login-ID-list* Remove any explicit priority level for the specified users.
- q** *priority-level* Set the default highest priority level for all users not explicitly covered.
- l** List the default priority level and the priority limits assigned to users.

SEE ALSO [lp\(1\)](#)

NAME	mail.local – store mail in a mailbox
SYNOPSIS	<code>/usr/lib/mail.local [-f sender] [-d] recipient</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>mail.local reads the standard input up to an end-of-file and appends it to each user's mail file (mailbox). This program is intended to be used by sendmail(1M) as a mail delivery agent for local mail. It is not a user interface agent.</p> <p>Messages are appended to the user's mail file in the <code>/var/mail</code> directory. The user must be a valid user name.</p> <p>Each delivered mail message in the mailbox are preceded by a "Unix From line" with the following format:</p> <p style="padding-left: 40px;">From sender_address time_stamp</p> <p>The sender_address is extracted from the From: header in the message. If the From: header is absent, the envelope address will be used. (the envelope address is specified with the <code>-f</code> option)</p> <p>A trailing blank line is also added to the end of each message.</p> <p>The mail files are locked with a .lock file while mail is appended.</p> <p>The mail files are created will mode 660, owner is set to recipient and group is set to mail. If the "biff" service is returned by getservbyname(3N), the biff server is notified of delivered mail. This program also computes the Content-Length: header which will be used by the mailbox reader to mark message boundary.</p>
OPTIONS	<p><code>-f sender</code> specified the "envelope from address" of the message This flag is optional and is mainly here for backward compatibility.</p> <p><code>-d</code> specified the recipient of the message, this flag is also optional and is supported here for backward compatibility. (i.e mail.local recipient is the same as mail.local -d recipient)</p>
ENVIRONMENT	TZ Used to set the appropriate time zone on the timestamp.
EXIT CODES	<p>0 on success.</p> <p>>0 if an error occurs.</p>
FILES	<p><code>/tmp/local.XXXXXXX</code> temporary files</p> <p><code>/tmp/lochd.XXXXXXX</code> temporary files</p> <p><code>/var/mail/user_name</code> user's mail file</p>
SEE ALSO	mail(1) , comsat(1M) , sendmail(1M) , getservbyname(3N)

NAME	makedbm – make a dbm file, or get a text file from a dbm file
SYNOPSIS	<pre> /usr/sbin/makedbm [-b] [-l] [-s] [-i yp_input_file] [-o yp_output_name] [-d yp_domain_name] [-m yp_master_name] infile outfile makedbm [-u dbmfilename] </pre>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>The makedbm command takes <i>infile</i> and converts it to a pair of files in ndbm(3) format, namely <i>outfile.pag</i> and <i>outfile.dir</i>. Each line of the input file is converted to a single dbm record. All characters up to the first TAB or SPACE form the key, and the rest of the line is the data. If a line ends with ‘ \ ’ (backslash), the data for that record is continued on to the next line. makedbm does not treat ‘ # ’ (pound-sign) as a special character. <i>infile</i> can be ‘ - ’ (dash), in which case the standard input is read.</p> <p>makedbm is mainly used in generating dbm files for the NIS name service, and, to that end, it generates a special entry with the key <i>yp_last_modified</i>, which is the date of <i>infile</i> (or the current time, if <i>infile</i> is ‘ - ’). The entries that have keys with the prefix <i>yp_</i> are interpreted by NIS server utilities.</p>
OPTIONS	<p>-b Insert the YP_INTERDOMAIN into the output. This key causes ypserv to use DNS for host name and address lookups for hosts not found in the maps.</p> <p>-l Lower case. Convert the keys of the given map to lower case, so that, for example, host name matches will succeed independent of upper or lower case distinctions.</p> <p>-s Secure map. Accept connections from secure NIS networks only.</p> <p>-i yp_input_file Create a special entry with the key <i>yp_input_file</i>.</p> <p>-o yp_output_name Create a special entry with the key <i>yp_output_name</i>.</p> <p>-d yp_domain_name Create a special entry with the key <i>yp_domain_name</i>.</p> <p>-m yp_master_name Create a special entry with the key <i>yp_master_name</i>. If no master host name is specified, <i>yp_master_name</i> will be set to the local host name.</p> <p>-u dbmfilename Undo a dbm file; i.e., print out the file in text format, one entry per line, with a single space separating keys from values.</p>
SEE ALSO	ndbm(3)

NAME	mk – remake the binary system and commands from source code
DESCRIPTION	<p>All source code for the UNIX system is distributed in the directory <code>/usr/src</code>. The directory tree rooted at <code>/usr/src</code> includes source code for the operating system, libraries, commands, miscellaneous data files necessary for the system and procedures to transform this source code into an executable system.</p> <p>Within the <code>/usr/src</code> directory are the <code>cmd</code>, <code>lib</code>, <code>uts</code>, <code>head</code>, and <code>stand</code> directories, as well as commands to remake the parts of the system found under each of these sub-directories. These commands are named <code>:mk</code> and <code>:mkdir</code> where <i>dir</i> is the name of the directory to be recreated. Each of these <code>:mkdir</code> commands rebuilds all or part of the directory it is responsible for. The <code>:mk</code> command runs each of the other commands in order and thus recreates the whole system. The <code>:mk</code> command is distributed only to source code licensees.</p> <p>Each command, with its associated directory, is described below.</p> <p>:mklib The <code>lib</code> directory contains the source code for the system libraries. The most important of these is the C library. Each library is in its own sub-directory. If any arguments are specified on the <code>:mklib</code> command line then only the given libraries are rebuilt. The argument <code>*</code> causes it to rebuild all libraries found under the <code>lib</code> directory.</p> <p>:mkhead The <code>head</code> directory contains the source code versions of the headers found in the <code>/usr/include</code> directory. The <code>:mkhead</code> command installs the headers given as arguments. The argument <code>*</code> causes it to install all headers.</p> <p>:mkuts The <code>uts</code> directory contains the source code for the UNIX Operating System. The <code>:mkuts</code> command takes no arguments and invokes a series of makefiles that recreates the operating system.</p> <p>Associated with the operating system is a set of headers that describe the user interface to the operating system. The source for these headers is found in a sub-directory within the <code>uts</code> directory tree. The user-accessible versions of these headers are found in the <code>/usr/include/sys</code> directory. The <code>:mksyshead</code> command installs these headers into the <code>/usr/include/sys</code> directory.</p> <p>:mkstand The <code>stand</code> directory contains stand-alone commands and boot programs. The <code>:mkstand</code> command rebuilds and installs these programs. Note that these stand-alone programs are only applicable to the DEC processors and are not built for any other machine.</p> <p>:mkcmd The <code>cmd</code> directory contains the source code for all the commands available on the system. There are two types of entries within the <code>cmd</code> directory: commands whose source code consists of only one file with one of the following suffixes: <code>.l</code>, <code>.y</code>, <code>.c</code>, <code>.s</code>, <code>.sh</code>, or a sub-directory that contains the multiple source files that comprise a particular command or subsystem.</p>

Each sub-directory is assumed to have a makefile (see **make(1S)**) with the name *command.mk* that takes care of creating everything associated with that directory and its sub-directories.

The **:mkcmd** command transforms source code into an executable command based on a set of predefined rules. If the **:mkcmd** command encounters a sub-directory within the **cmd** directory then it runs the makefile found in that sub-directory. If no makefile is found then an error is reported. For single-file commands, the predefined rules are dependent on the file's suffix. C programs (.c) are compiled by the C compiler and loaded stripped with shared text. Assembly language programs (.s) are assembled and loaded stripped. Yacc programs (.y) and lex programs (.l) are processed by **yacc()** and **lex()** respectively, before C compilation. Shell programs (.sh) are copied to create the command. Each of these operations leaves a command in the **./cmd** directory which is then installed into a user-accessible directory by using **/usr/sbin/install**.

The arguments to **:mkcmd** are either command names or subsystem names. Some subsystems distributed with the UNIX system are: **acct**, **graf**, **sgs**, **sccs**, and **text**. Prefacing the **:mkcmd** command with an assignment to the shell variable **\$ARGS** causes the indicated components of the subsystem to be rebuilt.

For example, the entire **sccs** subsystem can be rebuilt by:

```
/usr/src:/mkcmd sccs
```

while the **delta** component of **sccs** can be rebuilt by:

```
ARGS="delta" /usr/src:/mkcmd sccs
```

The **log** command, which is a part of the **stat** package, which is itself a part of the **graf** package, can be rebuilt by:

```
ARGS="stat log" /usr/src:/mkcmd graf
```

The argument ***** causes all commands and subsystems to be rebuilt.

Makefiles throughout the system, and particularly in the **cmd** directory, have a standard format. In particular, **:mkcmd** depends on each makefile having target entries for **install** and **clobber**. The **install** target should cause everything over which the makefile has jurisdiction to be built and installed by **/usr/sbin/install**. The **clobber** target should cause a complete cleanup of all unnecessary files resulting from the previous invocation. The commands that use the **CLOBBER** environment variable are **:mkcmd**, **:mklib**, and **:mkuts**. These commands all check the **CLOBBER** variable before executing **make clobber**. If this variable is set to **OFF**, then **make clobber** is *not* performed. If the variable is *not* set or is set to anything other than **OFF**, the **make clobber** is performed.

An effort has been made to separate the creation of a command from source and its installation on the running system. The command **/usr/sbin/install** is used by **:mkcmd** and most makefiles to install commands in standard directories on the system. The use of **install** allows maximum flexibility in the administration of the system. The **install** command makes very few assumptions about where a command is located, who owns it, and

what modes are in effect. All assumptions may be overridden on invocation of the command, or more permanently by redefining a few variables in **install**. The purpose of **install** is to install a new version of a command in the same place, with the same attributes as the prior version.

In addition, the use of a separate command to perform installation allows for the creation of test systems in other than standard places, easy movement of commands to balance load, and independent maintenance of makefiles.

SEE ALSO **install(1M)**, **make(1S)**

NAME	mkfifo – make FIFO special file
SYNOPSIS	<code>/usr/bin/mkfifo [-m <i>mode</i>] <i>path</i> ...</code>
AVAILABILITY	SUNWesu
DESCRIPTION	<p>The mkfifo command creates the FIFO special files named by its argument list. The arguments are taken sequentially, in the order specified; and each FIFO special file is either created completely or, in the case of an error or signal, not created at all.</p> <p>If errors are encountered in creating one of the special files, mkfifo writes a diagnostic message to the standard error and continues with the remaining arguments, if any.</p> <p>The mkfifo command calls the library routine mkfifo(3C), with the <i>path</i> argument is passed as the <i>path</i> argument from the command line, and <i>mode</i> is set to the equivalent of a=rw, modified by the current value of the file mode creation mask umask(1).</p>
OPTIONS	<p>The following option is supported:</p> <p>-m <i>mode</i> Set the file permission bits of the newly-created FIFO to the specified <i>mode</i> value. The <i>mode</i> option-argument will be the same as the <i>mode</i> operand defined for the chmod(1) command. In <i><symbolicmode></i> strings, the <i>op</i> characters + and – will be interpreted relative to an assumed initial mode of a=rw.</p>
OPERANDS	<p>The following operand is supported:</p> <p><i>file</i> A path name of the FIFO special file to be created.</p>
ENVIRONMENT	See environ(5) for descriptions of the following environment variables that affect the execution of mkfifo : LC_CTYPE , LC_MESSAGES , and NLSPATH .
EXIT STATUS	<p>The following exit values are returned:</p> <p>0 All the specified FIFO special files were created successfully.</p> <p>>0 An error occurred.</p>
SEE ALSO	mkfifo(3C) , environ(5)

NAME	mkfile – create a file
SYNOPSIS	mkfile [-nv] <i>size</i> [k b m] <i>filename</i> ...
AVAILABILITY	SUNWcsu
DESCRIPTION	mkfile creates one or more files that are suitable for use as NFS-mounted swap areas, or as local swap areas. The sticky bit is set, and the file is padded with zeroes by default. The default <i>size</i> is in bytes, but it can be flagged as kilobytes, blocks, or megabytes, with the k , b , or m suffixes, respectively.
OPTIONS	-n Create an empty <i>filename</i> . The size is noted, but disk blocks are not allocated until data is written to them. Files created with this option cannot be swapped over local UFS mounts. -v Verbose. Report the names and sizes of created files.
SEE ALSO	swap (1M)

NAME	mkfs – construct a file system
SYNOPSIS	mkfs [-F <i>FSType</i>] [<i>generic_options</i>] [-o <i>FSType-specific_options</i>] <i>raw_device_file</i> [<i>operands</i>]
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>mkfs constructs a file system on the the <i>raw_device_file</i> by calling the specific mkfs module indicated by -F <i>FSType</i>.</p> <p>Note: ufs file systems are normally created with the newfs(1M) command.</p> <p><i>generic_options</i> are independent of file system type. <i>FSType-specific_options</i> is a comma-separated list of <i>keyword=value</i> pairs (with no intervening spaces), which are <i>FSType</i>-specific. <i>raw_device_file</i> specifies the disk partition on which to write the file system. It is required and must be the first argument following the <i>specific_options</i> (if any). <i>operands</i> are <i>FSType</i>-specific. See the <i>FSType</i>-specific manual page of mkfs (for example, mkfs_ufs(1M)) for a detailed description.</p>
OPTIONS	<p>The following are the generic options for mkfs:</p> <p>-F Specify the <i>FSType</i> to be constructed. If -F is not specified, the <i>FSType</i> is determined from /etc/vfstab by matching the <i>raw_device_file</i> with a vfstab entry, or by consulting the /etc/default/fs file.</p> <p>-V Echo the complete command line, but do not execute the command. The command line is generated by using the options and arguments provided and adding to them information derived from /etc/vfstab or /etc/default/fs. This option may be used to verify and validate the command line.</p> <p>-m Return the command line which was used to create the file system. The file system must already exist. This option provides a means of determining the command used in constructing the file system.</p> <p>-o Specify <i>FSType</i>-specific options. See the manual page for the mkfs module specific to the file system type.</p>
FILES	<p>/etc/default/fs Default file system type. Default values can be set for the following flags in /etc/default/fs. For example: LOCAL=ufs</p> <p>LOCAL: The default partition for a command if no <i>FSType</i> is specified.</p> <p>/etc/vfstab List of default parameters for each file system</p>
SEE ALSO	newfs (1M), vfstab (4) Manual pages for the <i>FSType</i> -specific modules of mkfs .
NOTES	This command may not be supported for all <i>FSTypes</i> .

NAME	mkfs_ufs – construct a ufs file system
SYNOPSIS	mkfs -F ufs [<i>generic_options</i>] [-o <i>FSType_specific_options</i>] <i>raw_device_file</i> [<i>size</i>]
DESCRIPTION	<p>The ufs-specific module of mkfs builds a ufs file system with a root directory and a lost+found directory (see fsck(1M)).</p> <p>Note: The ufs-specific mkfs is rarely run directly; use the newfs(1M) command instead. <i>raw_device_file</i> indicates the disk partition to write on unless the -o N option has been specified, or either the -V or -m generic options are passed from the generic mkfs module. <i>size</i> specifies the number of sectors in the file system. This argument must follow the <i>raw_device_file</i> argument, and is required (even with -o N) unless the -V or -m generic options are specified.</p> <p><i>generic_options</i> are supported by the generic mkfs command. See mkfs(1M) for a description of these options.</p>
OPTIONS	<p>-o Use one or more of the following values separated by commas (with no intervening spaces) to specify ufs-specific options:</p> <p>N Print out the file system parameters without actually creating the file system.</p> <p>nsect=<i>n</i> The number of sectors per track on the disk. The default is 32.</p> <p>ntrack=<i>n</i> The number of tracks per cylinder on the disk. The default is 16.</p> <p>bsize=<i>n</i> Logical block size, either 4096 or 8192. The default is 8192.</p> <p>fragsize=<i>n</i> The smallest amount of disk space in bytes to allocate to a file. The value must be a power of 2 selected from the range 512 to the logical block size. If logical block size is 4096, legal values are 512, 1024, 2048 and 4096; if logical block size is 8192, 8192 is also a legal value. The default is 1024.</p> <p>cgsize=<i>n</i> The number of cylinders per cylinder group. The default is 16.</p> <p>free=<i>n</i> The minimum percentage of free space to maintain in the file system. This space is off-limits to normal users. Once the file system is filled to this threshold, only the superuser can continue writing to the file system. This parameter can be subsequently changed using the tunefs(1M) command. The default is 10%.</p> <p>rps=<i>n</i> The rotational speed of the disk, in revolutions per second. The default is 60.</p>

nbpi=<i>n</i>	The number of bytes per inode, which specifies the density of inodes in the file system. The number is divided into the total size of the file system to determine the fixed number of inodes to create. It should reflect the expected average size of files in the file system. If fewer inodes are desired, a larger number should be used; to create more inodes a smaller number should be given. The default is 2048 .
opt=<i>a</i>	Space or time optimization preference; s specifies optimization for space, t specifies optimization for time. The default is t . This parameter may be subsequently changed with the tunefs(1M) command.
apc=<i>n</i>	The number of alternates per cylinder to reserve for bad block replacement (SCSI devices only). The default is 0 .
gap=<i>n</i>	Rotational delay. The expected time (in milliseconds) to service a transfer completion interrupt and initiate a new transfer on the same disk. The value is used to decide how much rotational spacing to place between successive blocks in a file. This parameter can be subsequently changed using the tunefs(1M) command. The default is disk-type dependent.
nrpos=<i>n</i>	The number of different rotational positions in which to divide a cylinder group. The default is 8 .
maxcontig=<i>n</i>	The maximum number of blocks, belonging to one file, that will be allocated contiguously before inserting a rotational delay. For a 4K file system, the default is 14 ; for an 8K file system it is 7 . This parameter can be subsequently changed using the tunefs(1M) command. Note: This parameter also controls clustering. Regardless of the value of <i>gap</i> , clustering is enabled only when maxcontig is greater than 1. Clustering allows higher I/O rates for sequential I/O and is described in tunefs(1M) .

Alternatively, parameters can be entered as a list of space-separated values (without keywords) whose meaning is positional. In this case, the **-o** option is omitted and the list follows the size operand. This is the way **newfs** passes the parameters to **mkfs**.

SEE ALSO **fck(1M)**, **mkfs(1M)**, **newfs(1M)**, **tunefs(1M)**, **dir_ufs(4)**, **fs_ufs(4)**

DIAGNOSTICS

Warning: insufficient space in super block for rotational layout tables with *nsect sblock.fs_nsect* and *ntrak sblock.fs_ntrak*. (File system performance may be impaired.)

Occurs typically on very high density disks. On such disks, the file system structure cannot encode the proper disk layout information, resulting in suboptimal performance.

Warning: inode blocks/cyl group (*grp*) \geq data blocks (*num*) in last cylinder

User request for inodes/byte (with the **nbpi** keyword) and the disk geometry results in a situation in which the last truncated cylinder group can not contain the correct number of data blocks; some disk space is wasted.

Warning: *num* sector(s) in last cylinder group unallocated

User parameters and disk geometry conflict; some disk space is lost. A possible cause is the specified size being smaller than the partition size.

NAME	mknod – make a special file
SYNOPSIS	<code>/usr/sbin/mknod name [b c <i>major minor</i>] [p]</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	mknod makes a directory entry for a special file.
OPTIONS	b Indicates a block-type special file. c Indicates a character-type special file. p Used to create a FIFO (named pipe). <i>major</i> Specifies the <i>major</i> device number. <i>minor</i> Specifies the <i>minor</i> device number; can be either decimal or octal. The assignment of major device numbers is specific to each system. You must be the super-user to use this form of the command. <i>name</i> A special file to be created.
SEE ALSO	ftp(1) , in.ftpd(1M) , mknod(2) , symlink(2)
NOTES	If mknod() is used to create a device in a remote directory (Remote File Sharing), the major and minor device numbers are interpreted by the server. With the advent of physical device naming, it would be preferable to create a symbolic link to the physical name of the device (in the /devices subtree) rather than using mknod .

NAME	modinfo – display information about loaded kernel modules								
SYNOPSIS	<code>/usr/sbin/modinfo [-i <i>module-id</i>]</code>								
AVAILABILITY	SUNWcsu								
DESCRIPTION	<p>modinfo displays information about the loaded modules. The format of the information is as follows:</p> <p style="text-align: center;"><i>Id Loadaddr Size Info Rev Module Name</i></p> <p>where <i>Id</i> is the module ID, <i>Loadaddr</i> is the starting text address in hex, <i>size</i> is the size of text, data, and bss in hex bytes, <i>Info</i> is module specific info, <i>Rev</i> is the revision of the loadable modules system, and <i>Module Name</i> is the filename and description of the module. The module specific information is the block and character major numbers for drivers, the system call number for system calls, or, for other module types, the index into the appropriate kernel table:</p> <table border="0" style="margin-left: 40px;"> <tr> <td>fmodsw</td> <td>for streams modules</td> </tr> <tr> <td>vfssw</td> <td>for filesystems</td> </tr> <tr> <td>class</td> <td>for scheduling classes</td> </tr> <tr> <td>execsw</td> <td>for exec modules</td> </tr> </table>	fmodsw	for streams modules	vfssw	for filesystems	class	for scheduling classes	execsw	for exec modules
fmodsw	for streams modules								
vfssw	for filesystems								
class	for scheduling classes								
execsw	for exec modules								
OPTIONS	<code>-i <i>module-id</i></code> Display information about this module only.								
EXAMPLES	<p>The following example displays the status of module 2:</p> <pre>example% modinfo -i 2 Id Loadaddr Size Info Rev Module Name 2 ff08e000 1734 - 1 swapgeneric (root and swap configuration)</pre>								
SEE ALSO	modload(1M) , modunload(1M)								

NAME	modload – load a kernel module
SYNOPSIS	modload [-p] [-e <i>exec_file</i>] <i>filename</i>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>modload loads the loadable module <i>filename</i> into the running system. <i>filename</i> is an object file produced by ld -r. If <i>filename</i> is an absolute pathname then the file specified by that absolute path is loaded. If <i>filename</i> does not begin with a '/' then the path to load <i>filename</i> is relative to the current directory unless the -p option is specified. The kernel's modpath variable can be set using the /etc/system file. The default value of the kernel's modpath variable is set to the path where the operating system was loaded. Typically this is /kernel/usr/kernel. Hence if you type:</p> <p style="padding-left: 40px;">example# modload drv/foo</p> <p>Then the kernel will look for ./drv/foo.</p> <p>If you type:</p> <p style="padding-left: 40px;">example# modload -p drv/foo</p> <p>Then the kernel will look for /kernel/drv/foo and then /usr/kernel/drv/foo.</p>
OPTIONS	<p>-p Use the kernel's internal modpath variable as the search path for the module.</p> <p>-e <i>exec_file</i> Specify the name of a shell script or executable image file that is executed after the module is successfully loaded. The first argument passed is the module ID (in decimal). The other argument is module specific. The module specific information is: the block and character major numbers for drivers, the system call number for system calls, or, for other module types, the index into the appropriate kernel table. See modinfo(1M).</p>
SEE ALSO	<p>ld(1), add_drv(1M), kernel(1M), modinfo(1M), modunload(1M), system(4), modldrv(9S), modlinkage(9S), modlstrmod(9S), module_info(9S)</p> <p><i>Writing Device Drivers</i> <i>Solaris 1.x to Solaris 2.x Transition Guide</i></p>
NOTES	Use add_drv(1M) to add device drivers, not modload. See <i>Writing Device Drivers</i> for procedures on adding device drivers.

NAME	modunload – unload a module
SYNOPSIS	modunload <i>-i module_id</i> [<i>-e exec_file</i>]
AVAILABILITY	SUNWcsu
DESCRIPTION	modunload unloads a loadable module from the running system. The <i>module_id</i> is the ID of the module as shown by modinfo (1M). If ID is 0 , all modules that were autoloaded which are unloadable, are unloaded. Modules loaded by modload (1M) are not affected.
OPTIONS	<i>-i module_id</i> Specify the module to be unloaded. <i>-e exec_file</i> Specify the name of a shell script or executable image file to be executed before the module is unloaded. The first argument passed is the module id (in decimal). There are two additional arguments that are module specific. For loadable drivers, the second and third arguments are the block major and character major numbers respectively. For loadable system calls, the second argument is the system call number. For loadable exec classes, the second argument is the index into the execsw table. For loadable filesystems, the second argument is the index into the vfssw table. For loadable streams modules, the second argument is the index into the fmodsw table. For loadable scheduling classes, the second argument is the index into the class array. Minus one is passed for an argument that does not apply.
SEE ALSO	modinfo (1M), modload (1M)

NAME	monitor – SPARC system PROM monitor
SYNOPSIS	L1-A BREAK initial system power-on exit from a client program, e.g., the Operating System
AVAILABILITY	SPARC
DESCRIPTION	The CPU board of a workstation contains one or more EPROMs or EEPROMs. The program which executes from the PROMs is referred to as “the monitor”. Among other things, the monitor performs system initialization at power-on and provides a user interface.
Monitor Prompt	<p>The monitor of earlier workstations was known as the SunMON monitor and displayed the > for its prompt. See the SunMON MONITOR USAGE section for further details.</p> <p>Existing workstations use a monitor which is known as the OpenBoot monitor. The OpenBoot monitor typically displays ok as its prompt, but it may also display the > prompt under certain circumstances.</p> <p>If the 'auto-boot?' NVRAM parameter is set to 'false' when the workstation is powered on then the system will not attempt to boot and the monitor will issue its prompt. If 'auto-boot' is set to 'true' then the system will initiate the boot sequence. The boot sequence can be aborted by simultaneously pressing two keys on the system's keyboard: L1 and A (on older keyboards), or Stop and A (on newer keyboards). Note that either a lower case 'a' or an upper case 'A' will work for the keyboard abort sequence. If a console has been attached via one of the system's serial ports then the abort sequence can be accomplished by sending a BREAK – see the tip(1) manpage.</p> <p>When the NVRAM 'security-mode' parameter has been turned on, or when the value of the 'sunmon-compat?' parameter is true, then the OpenBoot monitor will display the message:</p> <p style="text-align: center;">Type b (boot), c (continue), or n (new command mode)</p> <p>and the > prompt will appear.</p>
OPENBOOT PROM USAGE	Some of the more useful commands that can be issued from OpenBoot's ok prompt are described here. Refer to the <i>OpenBoot Command Reference Manual</i> book for a complete list of commands.
Help	Help for various functional areas of the OpenBoot monitor can be obtained by typing help . The help listing will provide a number of other key words which can then be used in the help command to provide further details.
NVRAM Parameters	Each workstation contains one or more NVRAM devices which contains unique system ID information, as well as a set of user-configurable parameters. The NVRAM parameters allow the user a certain level of flexibility in configuring the system to act in a given

manner under a specific set of circumstances.

See the **eeeprom**(1M) manpage for a description of the parameters. This manpage also describes a way of setting the parameters from the OS level.

The following commands can be used at the OpenBoot monitor to access the NVRAM parameters.

printenv	Used to list the NVRAM parameters, along with their default values and current values.
setenv <i>pn pv</i>	Used to set or modify a parameter. The <i>pn</i> represents the parameter name, and <i>pv</i> represents the parameter value.
set-default <i>pn</i>	Used to set an individual parameter back to its default value.
set-defaults	Used to reset all parameters to their default values. (Note that ' set-defaults ' only affects parameters that have assigned default values.)

Hardware Checks and Diagnostics

The following commands are available for testing or checking the system's hardware. If the '**diag-switch?**' NVRAM parameter is set to true when the system is powered on, then a Power-On Self Test (POST) diagnostic will be run, if present, sending its results messages to the system's serial port A. Not all of the commands shown are available on all workstations.

test-all	Run the diagnostic tests on each device which has provided a self-test.
test floppy	Run diagnostics on the system's floppy device.
test /memory	Run the main memory tests. If the NVRAM parameter ' diag-switch? ' is set to true, then all of main memory is tested. If the parameter is false then only the amount of memory specified in the ' selftest-#megs ' NVRAM parameter will be tested.
test net	Test the network connection for the on-board network controller.
watch-net	Monitor the network attached to the on-board net controller.
watch-net-all	Monitor the network attached to the on-board net controller, as well as the network controllers installed in SBus slots.
watch-clock	Test the system's clock function.

System Information

The following commands are available for displaying information about the system. Not all commands are available on all workstations.

banner	Display the power-on banner.
.enet-addr	Display the system's Ethernet address.
.idprom	Display the formatted contents of the IDPROM.
module-info	Display information about the system's processor(s).
probe-scsi	Identify the devices attached to the on-board SCSI controller.
probe-scsi-all	Identify the devices attached to the on-board SCSI controller as well as those devices which are attached to SBus SCSI controllers.

show-disks	Display a list of the device paths for installed SCSI disk controllers.
show-displays	Display a list of the device paths for installed display devices.
show-nets	Display a list of the device paths for installed Ethernet controllers.
show-sbus	Display list of installed SBus devices.
show-tapes	Display a list of the device paths for installed SCSI tape controllers.
show-ttys	Display a list of the device paths for tty devices.
.traps	Display a list of the SPARC trap types.
.version	Display the version and date of the OpenBoot PROM.

Emergency Commands

These commands must be typed from the keyboard, they will not work from a console which is attached via the serial ports. With the exception of the Stop-A command, these commands are issued by pressing and holding down the indicated keys on the keyboard immediately after the system has been powered on. The keys must be held down until the monitor has checked their status. The Stop-A command can be issued at any time after the console display begins, and the keys do not need to be held down once they've been pressed. The Stop-D, Stop-F and Stop-N commands are not allowed when one of the security modes has been set. Not all commands are available on all workstations.

Stop	(L1)	Bypass the Power-On Self Test (POST). This is only effective if the system has been placed into the diagnostic mode.
Stop-A	(L1-A)	Abort the current operation and return to the monitor's default prompt.
Stop-D	(L1-D)	Set the system's ' diag-switch? ' NVRAM parameter to ' true ', which places the system in diagnostic mode. POST diagnostics, if present, will be run, and the messages will be displayed via the system's serial port A.
Stop-F	(L1-F)	Enter the OpenBoot monitor before the monitor has probed the system for devices. Issue the ' fexit ' command to continue with system initialization.
Stop-N	(L1-N)	Causes the NVRAM parameters to be reset to their default values. Note that not all parameters have default values.

Line Editor Commands

The following commands can be used while the monitor is displaying the **ok** prompt. Not all of these editing commands are available on all workstations.

CTRL-A	Place the cursor at the start of line.
CTRL-B	Move the cursor backward one character.
ESC-B	Move the cursor backward one word.
CTRL-D	Erase the character that the cursor is currently highlighting.
ESC-D	Erase the portion of word from the cursor's present position to the end of the word.
CTRL-E	Place the cursor at the end of line.
CTRL-F	Move the cursor forward one character.
ESC-F	Move the cursor forward one word.
CTRL-H	Erase the character preceding the cursor (also use Delete or Back Space)

ESC-H	Erase the portion of the word which precedes the cursor (use also CTRL-W)
CTRL-K	Erase from the cursor's present position to the end of the line.
CTRL-L	Show the command history list.
CTRL-N	Recall the next command from the command history list
CTRL-P	Recall a previous command from the command history list.
CTRL-Q	Quote the next character (used to type a control character).
CTRL-R	Retype the current line.
CTRL-U	Erase from the cursor's present position to the beginning of the line.
CTRL-Y	Insert the contents of the memory buffer into the line, in front (to the left) of the cursor.

nvrsrc The **nvrsrc** is an area of the system's NVRAM where users may store Forth programs. The programs which are stored in the **nvrsrc** will be executed each time the system is reset, provided that the '**use-nvrsrc?**' NVRAM parameter has been set to '**true**'. Refer to the *OpenBoot Command Reference Manual* book for information on how to edit and use the **nvrsrc**.

Restricted Monitor The command '**old-mode**' is used to move OpenBoot into a restricted monitor mode, causing the > prompt to be displayed. Only three commands are allowed while in the restricted monitor; the '**go**' command (to resume a program which was interrupted with the Stop-A command), the '**n**' command (to return to the normal OpenBoot monitor), and boot commands. The restricted monitor's boot commands will approximate the older **SunMON** monitor's boot command syntax. If a '**security-mode**' has been turned on then the restricted monitor becomes the default monitor environment. The restricted monitor may also become the default environment if the '**sunmon-compat?**' NVRAM parameter is set to true. (Note that not all workstations will have the '**sunmon-compat?**' parameter.)

SunMON PROM USAGE

The following commands are available systems with older **SunMON**-based PROM:

+ -	Increment or decrement the current address and display the contents of the new location.
^C <i>source destination n</i>	(caret-C) Copy, byte-by-byte, a block of length <i>n</i> from the <i>source</i> address to the <i>destination</i> address.
^I <i>program</i>	(caret-I) Display the compilation date and location of <i>program</i> .
^T <i>virtual_address</i>	(caret-T) Display the physical address to which <i>virtual_address</i> is mapped.
b [!] [<i>device</i> [<i>(c,u,p)</i>]] [<i>pathname</i>] [<i>arguments_list</i>]	
b [?]	Reset appropriate parts of the system and bootstrap a program. A ' ! ' (preceding the <i>device</i> argument) prevents the system reset from occurring. Programs can be loaded from various devices (such as a disk, tape, or Ethernet). ' b ' with no arguments will cause a default boot, either from a disk, or from an Ethernet controller.

'b?' displays all boot devices and their *devices*.

device one of

- le** Lance Ethernet
- ie** Intel Ethernet
- sd** SCSI disk, CDROM
- st** SCSI 1/4" or 1/2" tape
- fd** Diskette
- id** IPI disk
- mt** Tape Master 9-track 1/2" tape
- xd** Xylogics 7053 disk
- xt** Xylogics 1/2" tape
- xy** Xylogics 440/450 disk

c A controller number (**0** if only one controller),

u A unit number (**0** if only one driver), and

p A partition.

pathname A pathname for a program such as **/stand/diag**.

arguments_list

A list of up to seven arguments to pass to the program being booted.

c [*virtual_address*] Resume execution of a program. When given, *virtual_address* is the address at which execution will resume. The default is the current PC. Registers are restored to the values shown by the **d**, and **r** commands.

d [*window_number*] Display (dump) the state of the processor. The processor state is observable only after:

- An unexpected trap was encountered.
- A user program dropped into the monitor (by calling *abortent*).
- The user manually entered the monitor by typing L1-A or BREAK.

The display consists of the following:

- The special registers: PSR, PC, nPC, TBR, WIM, and Y
- Eight global registers
- 24 window registers (8 *in*, 8 *local*, and 8 *out*), corresponding to one of the 7 available windows. If a Floating-Point Unit is on board, its status register along with 32 floating-point registers are also shown.

window_number Display the indicated *window_number*, which can be any value between **0** and **6**, inclusive. If no window is specified and the PSR's current window pointer contains a valid window number, registers from the window that was active just prior to entry into the monitor are displayed. Otherwise, registers from window 0 are displayed.

e [*virtual_address*] [*action*] ...

Open the 16-bit word at *virtual_address* (default zero). The address is interpreted in the address space defined by the **s** command. See the **a** command for a description of *action*.

f *virtual_address1* *virtual_address2* *pattern* [*size*]

Fill the bytes, words, or long words from *virtual_address1* (lower) to *virtual_address2* (higher) with the constant, *pattern*. The *size* argument can take one of the following values:

b byte format (the default)

w word format

l long word format

For example, the following command fills the address block from **0x1000** to **0x2000** with the word pattern, **0xABCD**:

f 1000 2000 ABCD W

g [*vector*] [*argument*]

g [*virtual_address*] [*argument*]

Goto (jump to) a predetermined or default routine (first form), or to a user-specified routine (second form). The value of *argument* is passed to the routine. If the *vector* or *virtual_address* argument is omitted, the value in the PC is used as the address to jump to.

To set up a predetermined routine to jump to, a user program must, prior to executing the monitor's **g** command, set the variable ***romp->v_vector_cmd** to be equal to the virtual address of the desired routine. Predetermined routines need not necessarily return control to the monitor.

The default routine, defined by the monitor, prints the user-supplied *vector* according to the format supplied in *argument*. This format can be one of:

%x hexadecimal

%d decimal

g0

Force a panic and produce a crash dump when the monitor is running as a result of the system being interrupted,

g4

(Sun-4 systems only) Force a kernel stack trace when the monitor is running as a result of the system being interrupted,

- h** Display the help menu for monitor commands and their descriptions. To return to the monitor's basic command level, press ESCAPE or **q** before pressing RETURN.
- i** [*cache_data_offset*] [*action*]...
Modify cache data RAM command. Display and/or modify one or more of the cache data addresses. See the **a** command for a description of *action*.
- j** [*cache_tag_offset*] [*action*]...
Modify cache tag RAM command. Display and/or modify the contents of one or more of the cache tag addresses. See the **a** command for a description of *action*.
- k** [*reset_level*]
Reset the system, where *reset_level* is:
- 0** Reset VMEbus, interrupt registers, video monitor (Sun-4 systems). This is the default.
 - 1** Software reset.
 - 2** Power-on reset. Resets and clears the memory. Runs the EPROM-based diagnostic self test, which can take several minutes, depending upon how much memory is being tested.
- kb** Display the system banner.
- l** [*virtual_address*] [*action*]...
Open the long word (32 bit) at memory address *virtual_address* (default zero). The address is interpreted in the address space defined by the **s** command (below). See the **a** command for a description of *action*.
- m** [*virtual_address*] [*action*]...
Open the segment map entry that maps *virtual_address* (default zero). The address is interpreted in the address space defined by the **s** command. See the **a** command for a description of *action*.
- ne**
- ni** Disable, enable, or invalidate the cache, respectively.
- o** [*virtual_address*] [*action*]...
Open the byte location specified by *virtual_address* (default zero). The address is interpreted in the address space defined by the **s** command. See the **a** command for a description of *action*.
- p** [*virtual_address*] [*action*]...
Open the page map entry that maps *virtual_address* (default zero) in the address space defined by the **s** command. See the **a** command for a description of *action*.

q [*eprom_offset*] [*action*] . . .

Open the EEPROM *eprom_offset* (default zero) in the EEPROM address space. All addresses are referenced from the beginning or base of the EEPROM in physical address space, and a limit check is performed to insure that no address beyond the EEPROM physical space is accessed. This command is used to display or modify configuration parameters, such as: the amount of memory to test during self test, whether to display a standard or custom banner, if a serial port (A or B) is to be the system console, etc. See the **a** command for a description of *action*.

r [*register_number*]

r [*register_type*]

r [*w window_number*]

Display and/or modify one or more of the IU or FPU registers. A hexadecimal *register_number* can be one of:

0x00–0x0f	window(0,i0)–window(0,i7), window(0,i0)–window(0,i7)
0x16–0x1f	window(1,i0)–window(1,i7), window(1,i0)–window(1,i7)
0x20–0x2f	window(2,i0)–window(2,i7), window(2,i0)–window(2,i7)
0x30–0x3f	window(3,i0)–window(3,i7), window(3,i0)–window(3,i7)
0x40–0x4f	window(4,i0)–window(4,i7), window(4,i0)–window(4,i7)
0x50–0x5f	window(5,i0)–window(5,i7), window(5,i0)–window(5,i7)
0x60–0x6f	window(6,i0)–window(6,i7), window(6,i0)–window(6,i7)
0x70–0x77	g0, g1, g2, g3, g4, g5, g6, g7
0x78–0x7d	PSR, PC, nPC, WIM, TBR, Y.
0x7e–0x9e	FSR, f0–f31

Register numbers can only be displayed after an unexpected trap, a user program has entered the monitor using the *abortent* function, or the user has entered the monitor by manually typing L1–A or BREAK.

If a *register_type* is given, the first register of the indicated type is displayed. *register_type* can be one of:

f floating-point
g global
s special

If **w** and a *window_number* (**0–6**) are given, the first *in*-register within the indicated window is displayed. If *window_number* is

omitted, the window that was active just prior to entering the monitor is used. If the PSR's current window pointer is invalid, window 0 is used.

s [*asi*) Set or display the Address Space Identifier. With no argument, **s** displays the current Address Space Identifier. The *asi* value can be one of:

0x2	control space
0x3	segment table
0x4	Page table
0x8	user instruction
0x9	supervisor instruction
0xa	user data
0xb	supervisor data
0xc	flush segment
0xd	flush page
0xe	flush context
0xf	cache data

u [*echo*]

u [*port*] [*options*] [*baud_rate*]

u [**u**] [*virtual_address*]

With no arguments, display the current I/O device characteristics including: current input device, current output device, baud rates for serial ports A and B, an input-to-output echo indicator, and virtual addresses of mapped UART devices. With arguments, set or configure the current I/O device. With the **u** argument (**uu. . .**), set the I/O device to be the *virtual_address* of a UART device currently mapped.

echo Can be either **e** to enable input to be echoed to the output device, or **ne**, to indicate that input is not echoed.

port Assign the indicated *port* to be the current I/O device. *port* can be one of:

a	serial port A
b	serial port B
k	the workstation keyboard
s	the workstation screen

baud_rate Any legal baud rate.

options can be any combination of:

i	input
o	output
u	UART
e	echo input to output
ne	do not echo input

r reset indicated serial port (**a** and **b** ports only)

If either **a** or **b** is supplied, and no *options* are given, the serial port is assigned for both input and output. If **k** is supplied with no options, it is assigned for input only. If **s** is supplied with no options, it is assigned for output only.

v *virtual_address1 virtual_address2 [size]*

Display the contents of *virtual_address1* (lower) *virtual_address2* (higher) in the format specified by *size*:

b byte format (the default)

w word format

l long word format

Enter return to pause for viewing; enter another return character to resume the display. To terminate the display at any time, press the space bar.

For example, the following command displays the contents of virtual address space from address **0x1000** to **0x2000** in word format:

v 1000 2000 W

w [*virtual_address*] [*argument*]

Set the execution vector to a predetermined or default routine. Pass *virtual_address* and *argument* to that routine.

To set up a predetermined routine to jump to, a user program must, prior to executing the monitor's **w** command, set the variable ***romp->v_vector_cmd** to be equal to the virtual address of the desired routine. Predetermined routines need not necessarily return control to the monitor.

The default routine, defined by the monitor, prints the user-supplied *vector* according to the format supplied in *argument*. This format can be one of:

%x hexadecimal

%d decimal

x

Display a menu of extended tests. These diagnostics permit additional testing of such things as the I/O port connectors, video memory, workstation memory and keyboard, and boot device paths.

yc *context_number*

yp | **s** *context_number virtual_address*

Flush the indicated context, context page, or context segment.

c flush context *context_number*

p flush the page beginning at *virtual_address* within context *context_number*

s flush the segment beginning at *virtual_address* within

context *context_number*

SEE ALSO

tip(1), boot(1M), eeprom(1M)

OpenBoot Command Reference Manual

NAME	montbl – create monetary database
SYNOPSIS	montbl [-o <i>outfile</i>] <i>infile</i>
DESCRIPTION	<p>The montbl command takes as input a specification file, <i>infile</i>, that describes the formatting conventions for monetary quantities for a specific locale.</p> <p><i>infile</i> describes the monetary formatting conventions for a specific locale. The output of montbl is suitable for use by localeconv(3C), after the super-user has installed <i>outfile</i> as <i>/usr/lib/locale/locale/LC_MONETARY</i>. This file must be readable by everybody.</p> <p>Before calling localeconv(), a program should set the locale category LC_MONETARY using setlocale(3C). This file is used by the localeconv() function to initialize the monetary specific fields of a structure of type struct lconv. For a description of the fields in this structure, see localeconv(3C).</p> <pre> struct lconv { char *decimal_point; /* "" (zero length string) */ char *thousands_sep; /* "" (zero length string) */ char *grouping; /* "" */ char *int_curr_symbol; /* "" */ char *currency_symbol; /* "" */ char *mon_decimal_point; /* "" */ char *mon_thousands_sep; /* "" */ char *mon_grouping; /* "" */ char *positive_sign; /* "" */ char *negative_sign; /* "" */ char int_frac_digits; /* CHAR_MAX */ char frac_digits; /* CHAR_MAX */ char p_cs_precedes; /* CHAR_MAX */ char p_sep_by_space; /* CHAR_MAX */ char n_cs_precedes; /* CHAR_MAX */ char n_sep_by_space; /* CHAR_MAX */ char p_sign_posn; /* CHAR_MAX */ char n_sign_posn; /* CHAR_MAX */ }; </pre> <p>The specification file specifies the value of each struct lconv member, except for the first two members, <i>decimal_point</i> and <i>thousands_sep</i>, which are set by the LC_NUMERIC category of setlocale(3C). Each member's value is given on a line in the order defined in the lconv structure.</p> <p>Blank lines and comment lines starting with # are ignored. Characters in strings may be in octal or hex representation; for example, <code>\141</code> or <code>\x61</code> indicate the letter 'a'. If there is no specification line for a given structure member, the default 'C' locale value for is used (these values are shown as comments in the struct lconv definition above). Here is a specification file for Japan. Note that <code>\3</code> is in octal, as specified by ANSI C. The string <code>\xa1\xef</code> is a multibyte sequence for the Yen symbol '¥'.</p>

```

# first eight items have string values, remaining items have
# numeric values.
# grouping=
\3
# int_curr_symbol=
JPY
# currency_symbol=
\xa1\xef
# mon_decimal_point=
.
# mon_thousands_sep=
,
# mon_grouping=
\3
# positive_sign=

# negative_sign=
-
# int_frac_digits=
0
# frac_digits=
0
# p_cs_precedes=
1
# p_sep_by_space=
0
# n_cs_precedes=
1
# n_sep_by_space=
0
# p_sign_posn=
1
# n_sign_posn=
4

```

OPTIONS **-o *outfile*** Write output to *outfile*; otherwise, write output to LC_MONETARY.

FILES **/usr/lib/locale/locale/LC_MONETARY** LC_MONETARY database for *locale*

SEE ALSO **localeconv(3C), setlocale(3C)**

NOTES Do not change files under the C locale, as this could cause undefined or nonstandard behavior.

NAME	mount, umount – mount or unmount file systems and remote resources
SYNOPSIS	<pre> mount [-p -v] mount [-F <i>FSType</i>] [<i>generic_options</i>] [-o <i>specific_options</i>] [-O] <i>special</i> <i>mount_point</i> mount [-F <i>FSType</i>] [<i>generic_options</i>] [-o <i>specific_options</i>] [-O] <i>special mount_point</i> mount -a [-F <i>FSType</i>] [-V] [<i>current_options</i>] [-o <i>specific_options</i>] [<i>mount_point. . .</i>] umount [-V] [-o <i>specific_options</i>] <i>special</i> <i>mount_point</i> umount -a [-V] [-o <i>specific_options</i>] [<i>mount_point. . .</i>] </pre>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>mount attaches a file system to the file system hierarchy at the <i>mount_point</i>, which is the pathname of a directory. If <i>mount_point</i> has any contents prior to the mount operation, these are hidden until the file system is unmounted.</p> <p>umount unmounts a currently mounted file system, which may be specified either as a <i>mount_point</i> or as <i>special</i>, the device on which the file system resides.</p> <p>mount and umount maintain a table of mounted file systems in <i>/etc/mnttab</i>, which is described in <i>mnttab</i>(4). mount adds an entry to the mount table; umount removes an entry from the table.</p> <p>When invoked with both the <i>special</i> and <i>mount_point</i> arguments and the -F option, mount validates all arguments except for <i>special</i> and invokes the appropriate <i>FSType</i>-specific mount module. If invoked with no arguments, mount lists all the mounted file systems recorded in the mount table, <i>/etc/mnttab</i>. If invoked with a partial argument list (with only one of <i>special</i> or <i>mount_point</i>, or with both <i>special</i> or <i>mount_point</i> specified but not <i>FSType</i>), mount will search <i>/etc/vfstab</i> for an entry that will supply the missing arguments. If no entry is found, and the <i>special</i> argument starts with "/", the default local file system type specified in <i>/etc/default/fs</i> will be used. Otherwise the default remote file system type will be used. The default remote file system type is determined by the first entry in the <i>/etc/dfs/fstypes</i> file. After filling in missing arguments, mount will invoke the <i>FSType</i>-specific mount module.</p> <p>Only a super-user can mount or unmount file systems using mount and umount. However, any user can use mount to list mounted file systems and resources.</p>
OPTIONS	<p>-F <i>FSType</i> Used to specify the <i>FSType</i> on which to operate. The <i>FSType</i> must be specified or must be determinable from <i>/etc/vfstab</i>, or by consulting <i>/etc/default/fs</i> or <i>/etc/dfs/fstypes</i>.</p> <p>-a [<i>mount_points. . .</i>] Perform mount or umount operations in parallel, when possible.</p>

If mount points are not specified, **mount** will mount all file systems whose **/etc/vfstab** "mount at boot" field is "yes". If mount points are specified, then **/etc/vfstab** "mount at boot" field will be ignored.

If mount points are specified, **umount** will only unmount those mount points. If none is specified, then **umount** will attempt to unmount all filesystems in **/etc/mnttab**, with the exception of certain system required file systems: **/**, **/usr**, **/var**, **/proc**, **/dev/fd**, and **/tmp**.

- p** Print the list of mounted file systems in the **/etc/vfstab** format. Must be the only option specified.
- v** Print the list of mounted file systems in verbose format. Must be the only option specified.
- V** Echo the complete command line, but do not execute the command. **umount** generates a command line by using the options and arguments provided by the user and adding to them information derived from **/etc/mnttab**. This option should be used to verify and validate the command line.
- generic_options* Options that are commonly supported by most *FSType*-specific command modules. The following options are available:
 - m** Mount the file system without making an entry in **/etc/mnttab**.
 - r** Mount the file system read-only.
- o** Specify *FSType*-specific options in a comma separated (without spaces) list of suboptions and keyword-attribute pairs for interpretation by the *FSType*-specific module of the command. (See **mount_ufs**(1M))
- O** Overlay mount. Allow the file system to be mounted over an existing mount point, making the underlying file system inaccessible. If a mount is attempted on a pre-existing mount point without setting this flag, the mount will fail, producing the error "device busy".

FILES	/etc/mnttab mount table
	/etc/default/fs default local file system type. Default values can be set for the following flags in /etc/default/fs . For example: LOCAL=ufs LOCAL: The default partition for a command if no <i>FSType</i> is specified.
	/etc/vfstab list of default parameters for each file system.

SEE ALSO **mount_cachefs**(1M), **mount_hsf**s(1M), **mount_nfs**(1M), **mount_pcfs**(1M), **mount_tmpfs**(1M), **mount_ufs**(1M), **mountall**(1M), **umountall**(1M), **setmnt**(1M), **mnttab**(4), **vfstab**(4), **lofs**(7FS), **pcfs**(7FS)

NOTES

If the directory on which a file system is to be mounted is a symbolic link, the file system is mounted on the directory to which the symbolic link refers, rather than on top of the symbolic link itself.

NAME	mount_cachefs – mount CacheFS file systems
SYNOPSIS	mount -F cachefs [<i>generic_options</i>] -o backfstype= <i>file_system_type</i> [<i>other cacheFS options</i>] <i>special mount_point</i>
DESCRIPTION	The CacheFS-specific version of the mount command mounts a cached file system; if necessary, it NFS-mounts its back file system. It also provides a number of CacheFS-specific options for controlling the caching process. For more information regarding back file systems, refer to the <i>System Administration Guide, Volume I</i> .
OPTIONS	To mount a CacheFS file system, use the generic mount command with the -F option followed by the argument cachefs . The following generic mount options are available: <ul style="list-style-type: none"> -m Mount the file system without making an entry in the <i>/etc/mnttab</i> file. -O Overlay mount. Allows the filesystem to be mounted over an existing mount point, making the underlying filesystem inaccessible. If a mount is attempted on a pre-existing mount point without setting this flag, mount will fail with the error: mount -F cachefs: mount failed Device busy. -r Mount the file system read-only. <p>The following arguments to the -o option are specifically for CacheFS mounts. Use commas to separate multiple options. Note: the backfstype argument must be specified.</p> <p>backfstype=<i>file_system_type</i> The file system type of the back file system (for example, nfs).</p> <p>backpath=<i>path</i> Specifies where the back file system is already mounted. If this argument is not supplied, CacheFS determines a mount point for the back file system. The back file system must be read-only.</p> <p>cachedir=<i>directory</i> The name of the cache directory.</p> <p>cacheid=<i>ID</i> <i>ID</i> is a string specifying a particular instance of a cache. If you do not specify a cache ID, CacheFS will construct one.</p> <p>write-around non-shared Write modes for CacheFS. The write-around mode (the default) handles writes the same as NFS does; that is, writes are made to the back file system, and the affected file is purged from the cache. You can use the non-shared mode when you are sure that no one else will be writing to the cached file system. In this mode, all writes are made to both the front and the back file system, and the file remains in the cache.</p>

noconst	Disables cache consistency checking. By default, periodic consistency checking is enabled. Specify noconst only when you know that the back file system will not be modified. Trying to perform cache consistency check using cfsadmin -s will result in error. demandconst and noconst are mutually exclusive.
demandconst	Verifies cache consistency only when explicitly requested, rather than the periodic checking that is done by default. A consistency check is requested by using the -s option of the cfsadmin(1M) command. This option is useful for back file systems that change infrequently, for example, /usr/openwin . demandconst and noconst are mutually exclusive.
local-access	Causes the front file system to interpret the mode bits used for access checking instead of having the back file system verify access permissions. Do not use this argument with secure NFS.
purge	Purge any cached information for the specified file system.
rw ro	Read-write (default) or read-only.
suid nosuid	Allow (default) or disallow set-uid execution.
acregmin=<i>n</i>	Specifies that cached attributes are held for at least <i>n</i> seconds after file modification. After <i>n</i> seconds, CacheFS checks to see if the file modification time on the back file system has changed. If it has, all information about the file is purged from the cache and new data is retrieved from the back file system. The default value is 30 seconds.
acregmax=<i>n</i>	Specifies that cached attributes are held for no more than <i>n</i> seconds after file modification. After <i>n</i> seconds, all file information is purged from the cache. The default value is 30 seconds.
acdirmin=<i>n</i>	Specifies that cached attributes are held for at least <i>n</i> seconds after directory update. After <i>n</i> seconds, CacheFS checks to see if the directory modification time on the back file system has changed. If it has, all information about the directory is purged from the cache and new data is retrieved from the back file system. The default value is 30 seconds.
acdirmax=<i>n</i>	Specifies that cached attributes are held for no more than <i>n</i> seconds after directory update. After <i>n</i> seconds, all directory information is purged from the cache. The default value is 30 seconds.
actimeo=<i>n</i>	Sets acregmin , acregmax , acdirmin , and acdirmax to <i>n</i> .

EXAMPLES

The following example CacheFS-mounts the file system **server1:/user2**, which is already NFS-mounted on **/usr/abc** as **/xyz**.

```
example# mount -F cachefs -o backfstype=nfs,backpath=/usr/abc,
        cachedir=/cache1 server1:/user2 /xyz
```

The lines similar to the following appear in the **/etc/mnttab** file after the **mount** command is executed:

```
server1:/user2    /usr/abc    nfs
/usr/abc         /cache1/xyz  cachefs    backfstype=nfs
```

SEE ALSO

cfsadmin(1M), **fsck_cachefs(1M)**, **mount(1M)**
System Administration Guide, Volume I

NAME	mount_hfs – mount hfs file systems
SYNOPSIS	<p>mount -F hfs [<i>generic_options</i>] [-o <i>FSType-specific_options</i>] [-O] <i>special</i> <i>mount_point</i></p> <p>mount -F hfs [<i>generic_options</i>] [-o <i>FSType-specific_options</i>] [-O] <i>special mount_point</i></p>
DESCRIPTION	<p>mount attaches a High Sierra file system (hfs) to the file system hierarchy at the <i>mount_point</i>, which is the pathname of a directory. If <i>mount_point</i> has any contents prior to the mount operation, these are hidden until the file system is unmounted.</p> <p>If mount is invoked with <i>special</i> or <i>mount_point</i> as the only arguments, mount will search <i>/etc/vfstab</i> to fill in the missing arguments, including the <i>FSType-specific_options</i>; see mount(1M) for more details.</p> <p>If the file system being mounted contains Rock Ridge extensions, by default they will be used, enabling support of features not normally available under High Sierra file systems such as symbolic links, and special files.</p>
OPTIONS	<p><i>generic_options</i></p> <p>See mount(1M) for the list of supported options.</p> <p>-o Specify hfs file system specific options. If invalid options are specified, a warning message is printed and the invalid options are ignored. The following options are available:</p> <p>ro Mount the file system read-only. This option is required.</p> <p>nrr “no Rock Ridge”: if Rock Ridge extensions are present in the file system, ignore them; interpret it as a regular High Sierra file system.</p> <p>notraildot File names on High Sierra file systems consist of a proper name and an extension separated by a ‘.’ (dot) character. By default, the separating dot is always considered part of the file’s name for all file access operations, even if there is no extension present. Specifying notrail-dot makes it optional to specify the trailing dot to access a file whose name lacks an extension.</p> <p>Exceptions:</p> <p style="padding-left: 2em;">This option is effective only on file systems for which Rock Ridge extensions are not active, either because they are not present on the CD-ROM, or they are explicitly ignored via the nrr option. If Rock Ridge extensions are active, hfs quietly ignores this option.</p> <p>nomaplcse</p> <p style="padding-left: 2em;">File names on High Sierra cdroms with no Rock Ridge extensions present should be uppercase characters only. By default, hfs maps file names read from a non-Rock Ridge disk to all lowercase characters. nomaplcse turns off this mapping. The exceptions for</p>

notraildot discussed above apply to **nomapcase**.

nosuid By default the file system is mounted with **setuid** execution allowed. Specifying **nosuid** causes the file system to be mounted with **setuid** execution disallowed.

- O Overlay mount. Allow the file system to be mounted over an existing mount point, making the underlying file system inaccessible. If a mount is attempted on a pre-existing mount point without setting this flag, the mount will fail, producing the error "device busy".

FILES

/etc/mnttab	table of mounted file systems
/etc/vfstab	list of default parameters for each file system

SEE ALSO **mount(1M)**, **mountall(1M)**, **mount(2)**, **mnttab(4)**, **vfstab(4)**

NOTES If the directory on which a file system is to be mounted is a symbolic link, the file system is mounted on the directory to which the symbolic link refers, rather than on top of the symbolic link itself.

NAME	mount_nfs – mount remote NFS resources
SYNOPSIS	mount [-F nfs] [-r] [-m] [-o specific_options] [-O] <i>resource</i> <i>mount_point</i>
DESCRIPTION	<p>mount attaches a named <i>resource</i> to the file system hierarchy at the pathname location <i>mount_point</i>, which must already exist. If <i>mount_point</i> has any contents prior to the mount operation, the contents remain hidden until the <i>resource</i> is once again unmounted. If the resource is listed in the /etc/vfstab file, the command line can specify either <i>resource</i> or <i>mount_point</i>, and mount will consult /etc/vfstab for more information. If the -F option is omitted, mount takes the file system type from /etc/vfstab.</p> <p>mount maintains a table of mounted file systems in /etc/mnttab, described in mnttab(4).</p>
OPTIONS	<p>-r Mount the specified file system read-only.</p> <p>-m Do not append an entry to the /etc/mnttab table of mounted file systems</p> <p>-o specific_options Set file system specific options according to a comma-separated list chosen from words below.</p> <p>rw ro <i>resource</i> is mounted read-write or read-only. The default is rw.</p> <p>suid nosuid Setuid execution allowed or disallowed. The default is suid.</p> <p>remount If a file system is mounted read-only, remounts the file system read-write.</p> <p>bg fg If the first attempt fails, retry in the background, or, in the foreground. The default is fg.</p> <p>quota Enables quota(1M) to check whether the user is over quota on this file system; if the file system has quotas enabled on the server, quotas will still be checked for operations on this file system.</p> <p>noquota Prevent quota(1M) from checking whether the user exceeded the quota on this file system; if the file system has quotas enabled on the server, quotas will still be checked for operations on this file system.</p> <p>retry=<i>n</i> The number of times to retry the mount operation. The default is 10000.</p> <p>vers=<NFS version number> By default, the version of NFS protocol used between the client and the server is the highest one available on both systems. If the NFS server does not support NFS Version 3, then the NFS mount will use NFS Version 2</p> <p>proto=<netid> <i><netid></i> is a value of network_id field from entry in the /etc/netconfig file. By default, the transport protocol used for the NFS mount will be first available connection oriented transport</p>

	supported on both the client and the server. If no connection oriented transport is found, then the first available connectionless transport is used. This default behavior can be overridden with the proto=<netid> option.
port=<i>n</i>	The server IP port number. The default is NFS_PORT .
gripid	By default, the GID associated with a newly created file will obey the System V semantics; that is, the GID is set to the effective GID of the calling process. This behavior may be overridden on a per-directory basis by setting the set-GID bit of the parent directory; in this case, the GID of a newly created file is set to the GID of the parent directory (see open(2) and mkdir(2)). Files created on file systems that are mounted with the gripid option will obey BSD semantics independent of whether the set-GID bit of the parent directory is set; that is, the GID is unconditionally inherited from that of the parent directory.
rsize=<i>n</i>	Set the read buffer size to <i>n</i> bytes. The default value is 32768 .
wsize=<i>n</i>	Set the write buffer size to <i>n</i> bytes. The default value is 32768 .
timeo=<i>n</i>	Set the NFS timeout to <i>n</i> tenths of a second. The default value is 11 .
retrans=<i>n</i>	Set the number of NFS retransmissions to <i>n</i> . The default value is 5 .
soft hard	Return an error if the server does not respond, or continue the retry request until the server responds. The default value is hard .
intr nointr	Allow (do not allow) keyboard interrupts to kill a process that is hung while waiting for a response on a hard-mounted file system. The default is intr .
secure	Use DES authentication for NFS transactions.
posix	Request POSIX.1 semantics for the file system. Requires a mount version 2 mountd(1M) on the server.
kerberos	Use Kerberos authentication for NFS transactions.
noac	Suppress attribute caching.
acdirmax=<i>n</i>	Hold cached attributes for no more than <i>n</i> seconds after directory update. The default value is 60 .
acdirmin=<i>n</i>	Hold cached attributes for at least <i>n</i> seconds after directory update. The default value is 30 .
acregmax=<i>n</i>	Hold cached attributes for no more than <i>n</i> seconds after file modification. The default value is 60 .
acregmin=<i>n</i>	Hold cached attributes for at least <i>n</i> seconds after file modification. The default value is 3 .

actimeo=*n* Set *min* and *max* times for regular files and directories to *n* seconds.

- O Overlay mount. Allow the file system to be mounted over an existing mount point, making the underlying file system inaccessible. If a mount is attempted on a pre-existing mount point without setting this flag, the mount will fail, producing the error "device busy".

NFS FILE SYSTEMS

Background vs. Foreground

File systems mounted with the **bg** option indicate that **mount** is to retry in the background if the server's mount daemon (**mountd**(1M)) does not respond. **mount** retries the request up to the count specified in the **retry=*n*** option. Once the file system is mounted, each NFS request made in the kernel waits **timeo=*n*** tenths of a second for a response. If no response arrives, the time-out is multiplied by **2** and the request is retransmitted. When the number of retransmissions has reached the number specified in the **retrans=*n*** option, a file system mounted with the **soft** option returns an error on the request; one mounted with the **hard** option prints a warning message and continues to retry the request.

Hard vs. Soft

File systems that are mounted read-write or that contain executable files should always be mounted with the **hard** option. Applications using **soft** mounted file systems may incur unexpected I/O errors.

Authenticated Requests

The server may require authenticated NFS requests from the client. Either **secure** or **kerberos** authentication may be required.

File Attributes

To improve NFS read performance, files and file attributes are cached. File modification times get updated whenever a write occurs. However, file access times may be temporarily out-of-date until the cache gets refreshed.

The attribute cache retains file attributes on the client. Attributes for a file are assigned a time to be flushed. If the file is modified before the flush time, then the flush time is extended by the time since the last modification (under the assumption that files that changed recently are likely to change soon). There is a minimum and maximum flush time extension for regular files and for directories. Setting **actimeo=*n*** sets flush time to *n* seconds for both regular files and directories.

EXAMPLES

To mount an NFS file system:

```
example# mount serv:/usr/src /usr/src
```

To mount an NFS file system readonly with no suid privileges:

```
example# mount -r -o nosuid serv:/usr/src /usr/src
```

To mount an NFS file system over Version 2, with the UDP transport:

```
example# mount -o vers=2,proto=udp serv:/usr/src /usr/src
```

FILES

/etc/mnttab	table of mounted file systems
/etc/dfs/fstypes	default distributed file system type
/etc/vfstab	table of automatically mounted resources

SEE ALSO **mountall(1M)**, **mountd(1M)**, **quota(1M)**, **mkdir(2)**, **mount(2)**, **open(2)**, **umount(2)**, **mnttab(4)**

NOTES

A NFS server should not attempt to mount its own filesystems (see **lofs(7FS)**).

If the directory on which a file system is to be mounted is a symbolic link, the file system is mounted on *the directory to which the symbolic link refers*, rather than being mounted on top of the symbolic link itself.

SunOS 4.X used the **bioid** maintenance procedure to perform parallel read-ahead and write-behind on NFS clients. SunOS 5.X obsoleted **bioid** with multi-threaded processing, which transparently performs parallel read-ahead and write-behind.

NAME	mount_pcfs – mount pcfs file systems
SYNOPSIS	mount -F pcfs [<i>generic_options</i>] [-o <i>FSType-specific_options</i>] <i>special</i> <i>mount_point</i> mount -F pcfs [<i>generic_options</i>] [-o <i>FSType-specific_options</i>] <i>special mount_point</i>
AVAILABILITY	SUNWesu
DESCRIPTION	mount attaches an MS-DOS file system (pcfs) to the file system hierarchy at the <i>mount_point</i> , which is the pathname of a directory. If <i>mount_point</i> has any contents prior to the mount operation, these are hidden until the file system is unmounted. If mount is invoked with <i>special</i> or <i>mount_point</i> as the only arguments, mount will search /etc/vfstab to fill in the missing arguments, including the <i>FSType-specific_options</i> ; see mount(1M) for more details. The <i>special</i> argument can be one of two special device file types: <ul style="list-style-type: none"> • A floppy disk, such as /dev/diskette0 or /dev/diskette1. • A DOS logical drive on a hard disk expressed as <i>device-name:logical-drive</i>, where <i>device-name</i> specifies the special block device-file for the whole disk and <i>logical-drive</i> is either a drive letter (c through z) or a drive number (1 through 24). Examples are /dev/dsk/c0t0d0p0:c and /dev/dsk/c0t0d0p0:1. The <i>special</i> device file type must have a formatted MS-DOS file system with either a 12-bit or a 16-bit File Allocation Table. Regular or BIG-DOS (greater than 32 megabytes in size) partitions can be mounted.
OPTIONS	<i>generic_options</i> See mount(1M) for the list of supported options. -o Specify pcfs file system specific options. If invalid options are specified, a warning message is printed and the invalid options are ignored. The following options are available: rw ro Mount the file system read/write or read-only. The default is rw .
FILES	/etc/mnttab table of mounted file systems /etc/vfstab list of default parameters for each file system
SEE ALSO	mount(1M) , mountall(1M) , mount(2) , mnttab(4) , vfstab(4) , pcfs(7FS)
NOTES	If the directory on which a file system is to be mounted is a symbolic link, the file system is mounted on the directory to which the symbolic link refers, rather than on top of the symbolic link itself. pcfs is currently not NFS mountable. Trying to mount a pcfs file system through NFS will fail with an EACCES error.

NAME	mount_s5fs – mount s5 file systems
SYNOPSIS	mount -F s5fs [<i>generic_options</i>] [-o <i>FSType-specific_options</i>] <i>special</i> <i>mount_point</i> mount -F s5fs [<i>generic_options</i>] [-o <i>FSType-specific_options</i>] <i>special mount_point</i>
AVAILABILITY	x86 SUNWs53
DESCRIPTION	mount attaches a s5 file system (a System V file system used by PC versions of UNIX) to the file system hierarchy at the <i>mount_point</i> , which is the pathname of a directory. If <i>mount_point</i> has any contents prior to the mount operation, these are hidden until the file system is unmounted. If mount is invoked with <i>special</i> or <i>mount_point</i> as the only arguments, mount will search <i>/etc/vfstab</i> to fill in the missing arguments, including the <i>FSType-specific_options</i> . See mount (1M). If <i>special</i> and <i>mount_point</i> are specified without any <i>FSType-specific_options</i> , the default is rw .
OPTIONS	See mount (1M) for the list of supported <i>generic_options</i> . -o Specify s5 file system specific options in a comma-separated list with no intervening spaces. If invalid options are specified, a warning message is printed and the invalid options are ignored. The following options are available: f Fake an <i>/etc/mnttab</i> entry, but do not actually mount any file systems. Parameters are not verified. n Mount the file system without making an entry in <i>/etc/mnttab</i> . rw ro Read-write or read-only. Default is rw . remount Used in conjunction with rw . A file system mounted read-only can be <i>remounted</i> read-write. Fails if the file system is not currently mounted or if the file system is mounted rw .
FILES	<i>/etc/mnttab</i> table of mounted file systems <i>/etc/vfstab</i> list of default parameters for each file system
SEE ALSO	mount (1M), mountall (1M), mount (2), mnttab (4), vfstab (4)
NOTES	If the directory on which a file system is to be mounted is a symbolic link, the file system is mounted on the directory to which the symbolic link refers, rather than on top of the symbolic link itself.

NAME	mount_tmpfs – mount tmpfs file systems
SYNOPSIS	mount [-F tmpfs] [-o size=sz] [-O] <i>special mount_point</i>
DESCRIPTION	<p>tmpfs is a memory based file system which uses kernel resources relating to the VM system and page cache as a file system.</p> <p>mount attaches a tmpfs file system to the file system hierarchy at the pathname location <i>mount_point</i>, which must already exist. If <i>mount_point</i> has any contents prior to the mount operation, these remain hidden until the file system is once again unmounted.</p> <p>The <i>special</i> argument is usually specified as swap but is in fact disregarded and assumed to be the virtual memory resources within the system.</p>
OPTIONS	<p>-o size=sz The <i>sz</i> argument controls the size of this particular tmpfs file system. If the argument is has a 'k' suffix, the number will be interpreted as a number of kilobytes. An 'm' suffix will be interpreted as a number of megabytes. No suffix is interpreted as bytes. In all cases, the actual size of the file system is the number of bytes specified, rounded up to the physical pagesize of the system.</p> <p>-O Overlay mount. Allow the file system to be mounted over an existing mount point, making the underlying file system inaccessible. If a mount is attempted on a pre-existing mount point without setting this flag, the mount will fail, producing the error "device busy".</p>
FILES	/etc/mnttab table of mounted file systems
SEE ALSO	mount(1M), mkdir(2), mount(2), open(2), umount(2), mnttab(4), tmpfs(7FS)
NOTES	If the directory on which a file system is to be mounted is a symbolic link, the file system is mounted on the directory to which the symbolic link refers, rather than on top of the symbolic link itself.

NAME	mount_ufs – mount ufs file systems
SYNOPSIS	<p>mount -F ufs [<i>generic_options</i>] [-o <i>FSType-specific_options</i>] [-O] <i>special</i> <i>mount_point</i></p> <p>mount -F ufs [<i>generic_options</i>] [-o <i>FSType-specific_options</i>] [-O] <i>special</i> <i>mount_point</i></p>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>mount attaches a ufs file system to the file system hierarchy at the <i>mount_point</i>, which is the pathname of a directory. If <i>mount_point</i> has any contents prior to the mount operation, these are hidden until the file system is unmounted.</p> <p>If mount is invoked with <i>special</i> or <i>mount_point</i> as the only arguments, mount will search <i>/etc/vfstab</i> to fill in the missing arguments, including the <i>FSType-specific_options</i>. See mount(1M).</p> <p>If <i>special</i> and <i>mount_point</i> are specified without any <i>FSType-specific_options</i>, the default is rw.</p>
OPTIONS	<p>See mount(1M) for the list of supported <i>generic_options</i>.</p> <p>-o Specify ufs file system specific options in a comma-separated list with no intervening spaces. If invalid options are specified, a warning message is printed and the invalid options are ignored. The following options are available:</p> <p>onerror=action where <i>action</i> = panic lock umount repair. This option specifies the action that UFS should take to recover from an internal inconsistency on a file system. These cause a forced system shutdown, a file system lock to be applied to the file system, the file system to be forcibly unmounted, or an automatic fsck, respectively. The default is panic. The repair option downgrades to "panic" if the UFS-aware service daemon (ufsd) is not installed on the system. ufsd is a component of the unbundled Disk-Suite product.</p> <p>toosoon=number[s m h d w y] This option specifies the minimum time that must elapse between detection of inconsistencies on a file system. If an inconsistency is detected within this time period the system is forced to shut down. This prevents pathologic repairing of a file system which is damaged repeatedly. The optional unit key letter sets the units to be seconds, minutes, hours, days, weeks, or years, respectively. The default value is 1w (1 week).</p> <p>f Fake an <i>/etc/mnttab</i> entry, but do not actually mount any file systems. Parameters are not verified.</p>

- m** Mount the file system without making an entry in **/etc/mnttab**.
- quota** Quotas are turned on for the file system.
- rw | ro** Read-write or read-only. Default is **rw**.
- rq** Read-write with quotas turned on. Equivalent to **rw**, **quota**.
- nosuid** By default the file system is mounted with Setuid execution allowed. Specifying **nosuid** causes the file system to be mounted with setuid execution disallowed. **nosuid** can also be used to disallow setuid when mounting devices.
- remount** Used in conjunction with **rw**. A file system mounted read-only can be *remounted* read-write. Fails if the file system is not currently mounted or if the file system is mounted **rw**.
- intr | nointr** Allow(do not allow) keyboard interrupts to kill a process that is waiting for an operation on a locked file system. The default is **intr**.
- O** Overlay mount. Allow the file system to be mounted over an existing mount point, making the underlying file system inaccessible. If a mount is attempted on a pre-existing mount point without setting this flag, the mount will fail, producing the error "device busy".

FILES **/etc/mnttab** table of mounted file systems
/etc/vfstab list of default parameters for each file system

SEE ALSO **mount(1M)**, **mountall(1M)**, **mount(2)**, **mnttab(4)**, **vfstab(4)**

NOTES If the directory on which a file system is to be mounted is a symbolic link, the file system is mounted on the directory to which the symbolic link refers, rather than on top of the symbolic link itself.

NAME	mountall, umountall – mount, unmount multiple file systems
SYNOPSIS	mountall [-F <i>FSType</i>] [-l -r] [<i>file_system_table</i>] umountall [-k] [-s] [-F <i>FSType</i>] [-l -r] umountall [-k] [-s] [-h <i>host</i>]
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>mountall is used to mount file systems specified in a file system table. The file system table must be in vfstab(4) format. If no <i>file_system_table</i> is specified, /etc/vfstab will be used. If ‘-’ is specified as <i>file_system_table</i>, mountall will read the file system table from the standard input. mountall only mounts those file systems with the mount at boot field set to yes in the <i>file_system_table</i>.</p> <p>Each file system which has an fsckdev entry specified in the file system table will be checked using fsck(1M) in order to determine if it may be safely mounted. If the file system does not appear mountable, it is fixed using fsck before the mount is attempted. File systems with a ‘-’ entry in the fsckdev field will be mounted without first being checked.</p> <p>umountall causes all mounted file systems except root, /proc, /var, and /usr to be unmounted. If the <i>FSType</i> is specified, mountall and umountall limit their actions to the <i>FSType</i> specified.</p>
OPTIONS	<p>-F Specify the <i>FSType</i> of the file system to be mounted or unmounted.</p> <p>-h <i>host</i> Unmount all file systems listed in /etc/mnttab that are remote-mounted from <i>host</i>.</p> <p>-k Use the “fuser -k <mount point>” command. (See the fuser(1M) man page for details.) The -k option sends the SIGKILL signal to each process using the file. Since this option spawns kills for each process, the kill messages may not show up immediately.</p> <p>-l Limit the action to local file systems.</p> <p>-r Limit the action to remote file system types.</p> <p>-s Do not perform the umount operation in parallel.</p>
FILES	<p>/etc/mnttab mounted file system table.</p> <p>/etc/vfstab table of file system defaults.</p>
SEE ALSO	fsck(1M) , mount(1M) , mnttab(4) , vfstab(4)
DIAGNOSTICS	No messages are printed if the file systems are mountable and clean. Error and warning messages come from fsck(1M) and mount(1M) .

NAME	mountd – NFS mount request server
SYNOPSIS	/usr/lib/nfs/mountd
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>mountd is an RPC server that answers file system mount requests. It reads the file /etc/dfs/sharetab (described in sharetab(4)), to determine which file systems are available for mounting by which machines. It also provides information as to what file systems are mounted by which clients. This information can be printed using the dfmounts(1M) command.</p> <p>The mountd daemon is automatically invoked in run level 3. Only super-user can run the mountd daemon.</p>
FILES	/etc/dfs/sharetab shared file system table
SEE ALSO	dfmounts(1M) , sharetab(4)
NOTES	Some routines that compare hostnames use case-sensitive string comparisons; some do not. If an incoming request fails, verify that the case of the hostname in the file to be parsed matches the case of the hostname called for, and attempt the request again.

NAME	mpstat – report per-processor statistics																																
SYNOPSIS	<code>/usr/bin/mpstat [<i>interval</i> [<i>count</i>]]</code>																																
AVAILABILITY	SUNWcsu																																
DESCRIPTION	<p>mpstat reports per-processor statistics in tabular form. Each row of the table represents the activity of one processor. The first table summarizes all activity since boot; each subsequent table summarizes activity for the preceding interval. All values are rates (events per second) unless otherwise noted.</p> <p>mpstat reports the following information:</p> <table> <tr> <td>CPU</td> <td>processor ID</td> </tr> <tr> <td>minf</td> <td>minor faults</td> </tr> <tr> <td>mjf</td> <td>major faults</td> </tr> <tr> <td>xcal</td> <td>inter-processor cross-calls</td> </tr> <tr> <td>intr</td> <td>interrupts</td> </tr> <tr> <td>ithr</td> <td>interrupts as threads (not counting clock interrupt)</td> </tr> <tr> <td>csw</td> <td>context switches</td> </tr> <tr> <td>icsw</td> <td>involuntary context switches</td> </tr> <tr> <td>migr</td> <td>thread migrations (to another processor)</td> </tr> <tr> <td>smtx</td> <td>spins on mutexes (lock not acquired on first try)</td> </tr> <tr> <td>srw</td> <td>spins on readers/writer locks (lock not acquired on first try)</td> </tr> <tr> <td>syscl</td> <td>system calls</td> </tr> <tr> <td>usr</td> <td>percent user time</td> </tr> <tr> <td>sys</td> <td>percent system time</td> </tr> <tr> <td>wt</td> <td>percent wait time</td> </tr> <tr> <td>idl</td> <td>percent idle time</td> </tr> </table>	CPU	processor ID	minf	minor faults	mjf	major faults	xcal	inter-processor cross-calls	intr	interrupts	ithr	interrupts as threads (not counting clock interrupt)	csw	context switches	icsw	involuntary context switches	migr	thread migrations (to another processor)	smtx	spins on mutexes (lock not acquired on first try)	srw	spins on readers/writer locks (lock not acquired on first try)	syscl	system calls	usr	percent user time	sys	percent system time	wt	percent wait time	idl	percent idle time
CPU	processor ID																																
minf	minor faults																																
mjf	major faults																																
xcal	inter-processor cross-calls																																
intr	interrupts																																
ithr	interrupts as threads (not counting clock interrupt)																																
csw	context switches																																
icsw	involuntary context switches																																
migr	thread migrations (to another processor)																																
smtx	spins on mutexes (lock not acquired on first try)																																
srw	spins on readers/writer locks (lock not acquired on first try)																																
syscl	system calls																																
usr	percent user time																																
sys	percent system time																																
wt	percent wait time																																
idl	percent idle time																																
OPTIONS	<p><i>interval</i> Report once each <i>interval</i> seconds.</p> <p><i>count</i> Only print <i>count</i> reports.</p>																																
SEE ALSO	sar(1) , iostat(1M) , sar(1M) , vmstat(1M)																																

NAME	mvdire – move a directory
SYNOPSIS	<i>/usr/sbin/mvdire dirname name</i>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>mvdire moves directories within a file system. <i>dirname</i> must be a directory. If <i>name</i> does not exist, it will be created as a directory. If <i>name</i> does exist, and is a directory, <i>dirname</i> will be created as <i>name/dirname</i>. <i>dirname</i> and <i>name</i> may not be on the same path; that is, one may not be subordinate to the other. For example:</p> <p style="padding-left: 40px;">example% mvdire x/y x/z</p> <p>is legal, but</p> <p style="padding-left: 40px;">example% mvdire x/y x/y/z</p> <p>is not.</p>
SEE ALSO	mkdir(1) , mv(1)
WARNINGS	Only the super-user can use mvdire .

NAME	ncheck – generate a list of path names versus i-numbers
SYNOPSIS	ncheck [-F <i>FSType</i>] [-V] [<i>generic_options</i>] [-o <i>FSType-specific_options</i>] [<i>special ...</i>]
AVAILABILITY	SUNWcsu
DESCRIPTION	ncheck with no options generates a path-name versus i-number list of all files on <i>special</i> . If <i>special</i> is not specified on the command line the list is generated for all <i>specials</i> in /etc/vfstab which have a numeric fsckpass . <i>special</i> is a block special device on which the file system exists.
OPTIONS	<p>-F Specify the <i>FSType</i> on which to operate. The <i>FSType</i> should either be specified here or be determinable from /etc/vfstab by finding an entry in the table that has a numeric fsckpass field and an fsckdev that matches <i>special</i>.</p> <p>-V Echo the complete command line, but do not execute the command. The command line is generated by using the options and arguments provided by the user and adding to them information derived from /etc/vfstab. This option may be used to verify and validate the command line.</p> <p><i>generic_options</i> Options that are commonly supported by most <i>FSType</i>-specific command modules. The following options are available:</p> <p>-i <i>i-list</i> Limit the report to the files on the <i>i-list</i> that follows. The <i>i-list</i> must be separated by commas with no intervening spaces.</p> <p>-a Print the names “.” and “..”, which are ordinarily suppressed.</p> <p>-s Report only special files and files with set-user-ID mode. This option may be used to detect violations of security policy.</p> <p>-o Specify <i>FSType-specific_options</i> in a comma separated (without spaces) list of suboptions and keyword-attribute pairs for interpretation by the <i>FSType</i>-specific module of the command.</p>
FILES	/etc/vfstab list of default parameters for each file system
SEE ALSO	vfstab(4) Manual pages for the <i>FSType</i> -specific modules of ncheck
NOTES	This command may not be supported for all <i>FSTypes</i> .

NAME	ncheck_ufs – generate pathnames versus i-numbers for ufs file systems
SYNOPSIS	ncheck –F ufs [<i>generic_options</i>] [–o m] [<i>special ...</i>]
DESCRIPTION	ncheck –F ufs generates a pathname versus i-number list of files for the ufs file system residing on <i>special</i> . Names of directory files are followed by <i>/. .</i>
OPTIONS	See ncheck (1M) for the list of <i>generic_options</i> supported. –o Specify ufs file system specific options. The available option is: m Print mode information.
DIAGNOSTICS	When the file system structure is improper, ‘??’ denotes the “parent” of a parentless file and a pathname beginning with ‘. . .’ denotes a loop.
SEE ALSO	ff (1M), ncheck (1M)

NAME	nnd – get and set driver configuration parameters
SYNOPSIS	nnd [-set] <i>driver parameter</i> [<i>value</i>]
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>nnd gets and sets selected configuration parameters in some kernel drivers. Currently, nnd only supports the drivers that implement the TCP/IP Internet protocol family. Each driver chooses which parameters to make visible using nnd. Since these parameters are usually tightly coupled to the implementation, they are likely to change from release to release. Some parameters may be read-only.</p> <p>If the -set option is omitted, nnd queries the named <i>driver</i>, retrieves the value associated with the specified <i>parameter</i>, and prints it. If the -set option is given, nnd passes <i>value</i>, which must be specified, down to the named <i>driver</i> which assigns it to the named <i>parameter</i>.</p> <p>By convention, drivers that support nnd also support a special read-only <i>parameter</i> named “?” which can be used to list the parameters supported by the driver.</p>
EXAMPLES	<p>To see which parameters are supported by the TCP driver, use the following command:</p> <pre>example% nnd /dev/tcp \?</pre> <p>NOTE: The parameter name “?” may need to be escaped with a backslash to prevent its being interpreted as a shell meta character.</p> <p>The following command sets the value of the parameter ip_forwarding in the IP driver to zero. This disables IP packet forwarding.</p> <pre>example% nnd -set /dev/ip ip_forwarding 0</pre> <p>To view the current IP forwarding table, use the following command:</p> <pre>example% nnd /dev/ip ip_ire_status</pre>
SEE ALSO	ioctl(2) , arp(7P) , ip(7P) , tcp(7P) , udp(7P)
NOTES	<p>The parameters supported by each driver may change from release to release. Like programs that read /dev/kmem, user programs or shell scripts that execute nnd should be prepared for parameter names to change.</p> <p>The ioctl() command that nnd uses to communicate with drivers is likely to change in a future release. User programs should avoid making dependencies on it.</p> <p>The meanings of many nnd parameters make sense only if you understand how the driver is implemented.</p>

NAME	netstat – show network status
SYNOPSIS	<pre>netstat [-anv] [system] [core] netstat [-s -g -m -p -f address_family] [-P protocol] [-n] [system] [core] netstat -i -I interface [interval] [system] [core] netstat -r [-anv] [system] [core] netstat -M [-ns] [system] [core]</pre>
DESCRIPTION	<p>netstat displays the contents of various network-related data structures in various formats, depending on the options you select.</p> <p>The first form of the command displays a list of active sockets for each protocol. The second form selects one from among various other network data structures. The third form displays the routing table and the fourth form the multicast routing table.</p>
OPTIONS	<p>-a Show the state of all sockets and all routing table entries; normally sockets used by server processes are not shown and only interface, host, network and default routes are shown.</p> <p>-f address_family Limit statistics or address control block reports to those of the specified address_family, which can be one of: inet For the AF_INET address family unix For the AF_UNIX address family.</p> <p>-g Show the multicast group memberships for all interfaces.</p> <p>-i Show the state of the interfaces that are used for TCP/IP traffic. (See ifconfig(1M)).</p> <p>-m Show the STREAMS statistics.</p> <p>-n Show network addresses as numbers. netstat normally displays addresses as symbols. This option may be used with any of the display formats.</p> <p>-p Show the address resolution (ARP) tables.</p> <p>-r Show the routing tables.</p> <p>-s Show per-protocol statistics. When used with the -M option, show multicast routing statistics instead.</p> <p>-v Verbose. Show additional information for the sockets and the routing table.</p> <p>-I interface Show the state of a particular interface. <i>interface</i> can be any valid interface such as ie0 or le0.</p> <p>-M Show the multicast routing tables. When used with the -s option, show multicast routing statistics instead.</p>

-P protocol Limit display of statistics or state of all sockets to those applicable to *protocol*.

DISPLAYS

Active Sockets (First Form)

The display for each active socket shows the local and remote address, the send and receive queue sizes (in bytes), the send and receive windows (in bytes), and the internal state of the protocol.

The symbolic format normally used to display socket addresses is either:

hostname.port

when the name of the host is specified, or:

network.port

if a socket address specifies a network but no specific host.

The numeric host address or network number associated with the socket is used to look up the corresponding symbolic hostname or network name in the *hosts* or *networks* database.

If the network or hostname for an address is not known (or if the **-n** option is specified), the numerical network address is shown. Unspecified, or "wildcard", addresses and ports appear as "*". For more information regarding the Internet naming conventions, refer to **inet(7P)**.

TCP Sockets

The possible state values for TCP sockets are as follows:

CLOSED	Closed. The socket is not being used.
LISTEN	Listening for incoming connections.
SYN_SENT	Actively trying to establish connection.
SYN_RECEIVED	Initial synchronization of the connection under way.
ESTABLISHED	Connection has been established.
CLOSE_WAIT	Remote shut down; waiting for the socket to close.
FIN_WAIT_1	Socket closed; shutting down connection.
CLOSING	Closed, then remote shutdown; awaiting acknowledgement.
LAST_ACK	Remote shut down, then closed; awaiting acknowledgement.
FIN_WAIT_2	Socket closed; waiting for shutdown from remote.
TIME_WAIT	Wait after close for remote shutdown retransmission.

Network Data Structures (Second Form)

The form of the display depends upon which of the **-i**, **-g**, **-m**, **-p** or **-s** options you select. If you specify more than one of these options, **netstat** displays the information for each one of them.

Routing Table (Third Form)

The routing table display lists the available routes and the status of each. Each route consists of a destination host or network, and a gateway to use in forwarding packets. The *flags* column shows the status of the route (**U** if "up"), whether the route is to a gateway (**G**), and whether the route was created dynamically by a redirect (**D**). If the **-a** option is specified there will be routing entries with flags for combined routing and address

resolution entries (**A**), broadcast addresses (**B**), and the local addresses for the host (**L**).

Interface routes are created for each interface attached to the local host; the gateway field for such entries shows the address of the outgoing interface.

The **refcnt** column gives the current number of routes that share the same link layer address.

The **use** column displays the number of packets sent using an combined routing and address resolution (**A**) or a broadcast (**B**) route. For a local (**L**) route this count is the number of packets received, and for all other routes it is the number of times the routing entry has been used to create a new combined route and address resolution entry.

The *interface* entry indicates the network interface utilized for the route.

**Multicast Routing
Tables (Fourth Form)**

The multicast routing table consists of the virtual interface table and the actual routing table.

SEE ALSO

ifconfig(1M), **iostat(1M)**, **vmstat(1M)**, **hosts(4)**, **networks(4)**, **protocols(4)**, **services(4)**

NOTES

The kernel's tables can change while **netstat** is examining them, creating incorrect or partial displays.

NAME	newfs – construct a new UFS file system
SYNOPSIS	newfs [-Nv] [<i>mkfs-options</i>] <i>raw-device</i>
DESCRIPTION	<p>newfs is a “friendly” front-end to the mkfs(1M) program for making UFS file systems on disk partitions. newfs calculates the appropriate parameters to use and calls mkfs.</p> <p><i>raw-device</i> is the name of a raw special device residing in /dev/rdisk (for example, /dev/rdisk/c0t0d0s6) on which to create the file system.</p> <p>If you run newfs interactively (that is, standard input is a tty), newfs will prompt for confirmation before making the file system.</p> <p>If the -N option is not specified and the inodes of the device are not randomized, newfs will call fsirand(1M).</p> <p>Note: You must be super-user to use this command.</p>
OPTIONS	<p>-N Print out the file system parameters that would be used in creating the file system without actually creating the file system. fsirand(1M) is not called here.</p> <p>-v Verbose. newfs prints out its actions, including the parameters passed to mkfs.</p> <p><i>mkfs-options</i></p> <p>Options that override the default parameters are:</p> <p>-a <i>apc</i> The number of alternate blocks per cylinder (SCSI devices only) to reserve for bad block replacement. The default is 0.</p> <p>-b <i>bsize</i> The logical block size of the file system in bytes (either 4096 or 8192). The default is 8192.</p> <p>-c <i>cgize</i> The number of cylinders per cylinder group (ranging from 1 to 32). The default is 16.</p> <p>-d <i>gap</i> Rotational delay. The expected time (in milliseconds) to service a transfer completion interrupt and initiate a new transfer on the same disk. It is used to decide how much rotational spacing to place between successive blocks in a file. This parameter can be subsequently changed using the tunefs(1M) command. The default is disk-type dependent.</p> <p>-f <i>fragsize</i> The smallest amount of disk space in bytes to allocate to a file. The values must be a power of two selected from the range 512 to the logical block size. If logical block size is 4096, legal values are 512, 1024, 2048 and 4096; if logical block size is 8192, 8192 is also a legal value. The default is 1024.</p> <p>-i <i>nbpi</i> The number of bytes per inode. This specifies the density of inodes in the file system. The number is divided into the total size of the file system to determine the fixed number of inodes to create. It should reflect the expected average size of files in the file system. If fewer inodes are desired, a larger number</p>

- should be used; to create more inodes a smaller number should be given. The default is **2048**.
- m free** The minimum percentage of free space to maintain in the file system. This space is off-limits to normal users. Once the file system is filled to this threshold, only the super-user can continue writing to the file system. This parameter can be subsequently changed using the **tunefs(1M)** command. The default is **10%**.
- n nrpos** The number of different rotational positions in which to divide a cylinder group. The default is **8**.
- o opt** (**space** or **time**). The file system can either be instructed to try to minimize the time spent allocating blocks, or to try to minimize the space fragmentation on the disk. If the minimum free space threshold (as specified by the **-m** option) is less than 10%, space optimization will be used.
- r rpm** The speed of the disk in revolutions per minute. The default is **3600**.
- s size** The size of the file system in sectors. The default is to use the entire partition.
- t ntrack** The number of tracks per cylinders on the disk. The default is taken from the disk label.
- C maxcontig** The maximum number of blocks, belonging to one file, that will be allocated contiguously before inserting a rotational delay. For a 4K file system, the default is **14**; for an 8K file system, the default is **7**. This parameter can be subsequently changed using the **tunefs(1M)** command.
- Note:** This parameter also controls clustering. Regardless of the value of *gap*, clustering is enabled only when *maxcontig* is greater than 1. Clustering allows higher I/O rates for sequential I/O and is described in **tunefs(1M)**.

EXAMPLES

The following example verbosely displays the parameters for the raw special device, **c0t0d0s6**, but does not actually create a new file system:

```
example# newfs -Nv /dev/rdisk/c0t0d0s6
mkfs -F ufs -o N /dev/rdisk/c0t0d0s6 1112940 54 15 8192 1024 16 10 60 2048 t 0 -1 8
/dev/rdisk/c0t0d0s6: 1112940 sectors in 1374 cylinders of 15 tracks, 54 sectors
569.8MB in 86 cyl groups (16 c/g, 6.64MB/g, 3072 i/g)
super-block backups (for fsck -b #) at:
32, 13056, 26080, 39104, 52128, 65152, 78176, 91200, 104224,
...
example#
```

SEE ALSO

fsck(1M), fsck_ufs(1M), fsirand(1M), mkfs(1M), mkfs_ufs(1M), tuneufs(1M), fs_ufs(4)

DIAGNOSTICS

newfs: special No such file or directory

The device specified does not exist, or a disk partition was not specified.

special: cannot open

You must be super-user to use this command.

NAME	newkey – create a new Diffie-Hellman key pair in the publickey database
SYNOPSIS	newkey -h <i>hostname</i> [-s nisplus nis files] newkey -u <i>username</i> [-s nisplus nis files]
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>newkey establishes new public keys for users and machines on the network. These keys are needed when using secure RPC or secure NFS service.</p> <p>newkey prompts for a password for the given <i>username</i> or <i>hostname</i> and then creates a new public/secret Diffie-Hellman 192 bit key pair for the user or host. The secret key is encrypted with the given password. The key pair can be stored in the /etc/publickey file, the NIS publickey map, or the NIS+ cred.org_dir table.</p> <p>newkey consults the publickey entry in the name service switch configuration file (see nsswitch.conf(4)) to determine which naming service is used to store the secure RPC keys. If the publickey entry specifies a unique name service, newkey will add the key in the specified name service. However, if there are multiple name services listed, newkey cannot decide which source to update and will display an error message. The user is required to specify the source explicitly with the -s option.</p> <p>In the case of NIS, newkey should be run by the superuser on the master NIS server for that domain. In the case of NIS+, newkey should be run by the superuser on a machine which has permission to update the cred.org_dir table of the new user/host domain.</p> <p>In the case of NIS+, nisaddcred(1M) should be used to add new keys.</p>
OPTIONS	<p>-h <i>hostname</i> Create a new public/secret key pair for the privileged user at the given <i>hostname</i>. Prompts for a password for the given <i>hostname</i>.</p> <p>-u <i>username</i> Create a new public/secret key pair for the given <i>username</i>. Prompts for a password for the given <i>username</i>.</p> <p>-s nisplus -s nis -s files Update the database in the specified source: nisplus (for NIS+), nis (for NIS), or files. Other sources may be available in the future.</p>
SEE ALSO	chkey(1) , keylogin(1) , nisaddcred(1M) , nisclient(1M) , nsswitch.conf(4) , publickey(4)

NAME	nfsd – NFS daemon
SYNOPSIS	nfsd [-a] [-p protocol] [-t device] [-c #_conn] [<i>nservers</i>]
DESCRIPTION	<p>nfsd is the daemon that handles client file system requests. Only the super-user can run this daemon.</p> <p>The nfsd daemon is automatically invoked in run level 3 with the -a option.</p> <p>By default nfsd will start over the tcp and udp transports.</p>
OPTIONS	<p>-a Start a NFS daemon over all available connectionless and connection-oriented transports, including udp and tcp.</p> <p>-p protocol Start a NFS daemon over the specified protocol.</p> <p>-t device Start a NFS daemon for the transport specified by the given device.</p> <p>-c #_conn This sets the maximum number of connections allowed to the NFS server over connection-oriented transports. By default, the number of connections is unlimited.</p> <p><i>nservers</i> This sets the maximum number of concurrent NFS requests that the server can handle. This concurrency is achieved by up to <i>nservers</i> threads created as needed in the kernel. <i>nservers</i> should be based on the load expected on this server. Sixteen is the usual number of <i>nservers</i>. If <i>nservers</i> is not specified, the maximum number of concurrent NFS requests will default to one.</p> <p>If the NFS_PORTMON variable is set, then clients are required to use privileged ports (ports < IPPORT_RESERVED) in order to get NFS services. This variable is equal to zero by default. This variable has been moved from the “nfs” module to the “nfssrv” module. To set the variable, edit the /etc/system file and add this entry:</p> <p style="text-align: center;">set nfssrv:nfs_portmon = 1</p>
FILES	.nfsXXX client machine pointer to an open-but-unlinked file
SEE ALSO	ps(1) , mountd(1M) , sharetab(4)
BUGS	The NFS service uses kernel threads to process all of the NFS requests. Currently, system utilization associated with these threads is not charged to the nfsd process. Therefore, ps(1) can report 0 cpu time associated with the NFS daemon, even though NFS processing is taking place on the server.

NAME	nfsstat – NFS statistics												
SYNOPSIS	nfsstat [-cmnrsz]												
AVAILABILITY	SUNWcsu												
DESCRIPTION	<p>nfsstat displays statistical information about the NFS and RPC (Remote Procedure Call), interfaces to the kernel. It can also be used to reinitialize this information. If no options are given the default is</p> <p style="text-align: center;">nfsstat -cnrs</p> <p>That is, display everything, but reinitialize nothing.</p>												
OPTIONS	<p>-c Display client information. Only the client side NFS and RPC information will be printed. Can be combined with the -n and -r options to print client NFS or client RPC information only.</p> <p>-m Display statistics for each NFS mounted file system. This includes the server name and address, mount flags, current read and write sizes, the retransmission count, and the timers used for dynamic retransmission. The srtt value contains the smoothed round trip time, the dev value contains the estimated deviation, and the cur value is the current backed-off retransmission value.</p> <p>-n Display NFS information. NFS information for both the client and server side will be printed. Can be combined with the -c and -s options to print client or server NFS information only.</p> <p>-r Display RPC information.</p> <p>-s Display server information.</p> <p>-z Zero (reinitialize) statistics. This option is for use by the super-user only, and can be combined with any of the above options to zero particular sets of statistics after printing them.</p>												
DISPLAYS	<p>The server RPC display includes the following fields:</p> <table border="0"> <tr> <td style="padding-right: 20px;">calls</td> <td>The total number of RPC calls received.</td> </tr> <tr> <td>badcalls</td> <td>The total number of calls rejected by the RPC layer (the sum of badlen and xdr call as defined below).</td> </tr> <tr> <td>nullrecv</td> <td>The number of times an RPC call was not available when it was thought to be received.</td> </tr> <tr> <td>badlen</td> <td>The number of RPC calls with a length shorter than a minimum-sized RPC call.</td> </tr> <tr> <td>xdr call</td> <td>The number of RPC calls whose header could not be XDR decoded.</td> </tr> <tr> <td>dupchecks</td> <td>The number of RPC calls that looked up in the duplicate request cache.</td> </tr> </table>	calls	The total number of RPC calls received.	badcalls	The total number of calls rejected by the RPC layer (the sum of badlen and xdr call as defined below).	nullrecv	The number of times an RPC call was not available when it was thought to be received.	badlen	The number of RPC calls with a length shorter than a minimum-sized RPC call.	xdr call	The number of RPC calls whose header could not be XDR decoded.	dupchecks	The number of RPC calls that looked up in the duplicate request cache.
calls	The total number of RPC calls received.												
badcalls	The total number of calls rejected by the RPC layer (the sum of badlen and xdr call as defined below).												
nullrecv	The number of times an RPC call was not available when it was thought to be received.												
badlen	The number of RPC calls with a length shorter than a minimum-sized RPC call.												
xdr call	The number of RPC calls whose header could not be XDR decoded.												
dupchecks	The number of RPC calls that looked up in the duplicate request cache.												

dupreqs The number of RPC calls that were found to be duplicates.

The server NFS display shows the number of NFS calls received (**calls**) and rejected (**bad-calls**), and the counts and percentages for the various calls that were made.

The client RPC display includes the following fields:

calls The total number of RPC calls made.

badcalls The total number of calls rejected by the RPC layer.

badxids The number of times a reply from a server was received which did not correspond to any outstanding call.

timeouts The number of times a call timed out while waiting for a reply from the server.

newcreds The number of times authentication information had to be refreshed.

badverfs The number of times the call failed due to a bad verifier in the response.

timers The number of times the calculated time-out value was greater than or equal to the minimum specified time-out value for a call.

cantconn The number of times the call failed due to a failure to make a connection to the server.

nomem The number of times the call failed due to a failure to allocate memory.

interrupts The number of times the call was interrupted by a signal before completing.

retrans The number of times a call had to be retransmitted due to a timeout while waiting for a reply from the server. Applicable only to RPC over connection-less transports.

The client NFS display shows the number of calls sent and rejected, as well as the number of times a **CLIENT** handle was received (**clgets**), the number of times the **CLIENT** handle cache had no unused entries (**cltoomany**), as well as a count of the various calls and their respective percentages.

SEE ALSO

SPARC: Installing Solaris Software
x86: Installing Solaris Software

NAME	nis_cachemgr – NIS+ utility to cache location information about NIS+ servers
SYNOPSIS	/usr/sbin/nis_cachemgr [-i] [-n] [-v]
AVAILABILITY	SUNWnisu
DESCRIPTION	<p>The nis_cachemgr daemon maintains a cache of the NIS+ directory objects. The cache contains location information necessary to contact the NIS+ servers that serve the various directories in the name space. This includes transport addresses, information needed to authenticate the server, and a time to live field which gives a hint on how long the directory object can be cached. The cache helps to improve the performance of the clients that are traversing the NIS+ name space. nis_cachemgr should be running on all the machines that are using NIS+. However, it is not required that the nis_cachemgr program be running in order for NIS+ requests to be serviced.</p> <p>The cache maintained by this program is shared by all the processes which access NIS+ on that machine. The cache is maintained in a file that is memory mapped (see mmap(2)) by all the processes. On start up, nis_cachemgr initializes the cache from the cold start file (see nisinit(1M)) and preserves unexpired entries that already exist in the cache file. Thus, the cache survives machine reboots.</p> <p>The nis_cachemgr program is normally started from a system startup script.</p> <p>Note: The nis_cachemgr program makes NIS+ requests under the NIS+ principal name of the host on which it runs. Before running nis_cachemgr, security credentials for the host should be added to the cred.org_dir table in the host's domain using nisaddcred(1M). Credentials of type DES will be needed if the NIS+ service is operating at security level 2 (see rpc.nisd(1M)). See the WARNINGS section, below. Additionally, a "keylogin -r" needs to be done on the machine.</p> <p>nisshowcache(1M) can be used to look at the cached objects.</p>
OPTIONS	<p>-i Force nis_cachemgr to ignore the previous cache file and reinitialize the cache from just the cold start file. By default, the cache manager initializes itself from both the cold start file and the old cache file, thereby maintaining the entries in the cache across machine reboots.</p> <p>-n Run nis_cachemgr in an <i>insecure</i> mode. By default, before adding a directory object to the shared cache, on the request of another process on the machine, it checks the encrypted signature on the request to make sure that the directory object is a valid one and is sent by an authorized server. In this mode, nis_cachemgr adds the directory object to the shared cache without making this check.</p> <p>-v This flag sets <i>verbose</i> mode. In this mode, the nis_cachemgr program logs not only errors and warnings, but also additional status messages. The additional messages are logged using syslog(3) with a priority of LOG_INFO.</p>

FILES	/var/nis/NIS_SHARED_DIRCACHE the shared cache file /var/nis/NIS_COLD_START the coldstart file /etc/init.d/rpc initialization scripts for NIS+
SEE ALSO	keylogin(1) , nisaddcred(1M) , nisinit(1M) , nisshowcache(1M) , rpc.nisd(1M) , mmap(2) , syslog(3) , nisfiles(4)
DIAGNOSTICS	The nis_cachemgr daemon logs error messages and warnings using syslog (see syslog(3)). Error messages are logged to the DAEMON facility with a priority of LOG_ERR , and warning messages with a priority of LOG_WARNING . Additional status messages can be obtained using the -v option.
WARNINGS	If the host principal does not have the proper security credentials in the cred.org_dir table for its domain, then running this program without the '-n' insecure mode option may significantly <i>degrade</i> the performance of processes issuing NIS+ requests.

NAME	<code>nisaddcred</code> – create NIS+ credentials															
SYNOPSIS	<pre><code>nisaddcred [-p principal] [-P nis_principal] [-l login_password] auth_type [domain_name] nisaddcred -r [nis_principal] [domain_name]</code></pre>															
AVAILABILITY	SUNWnisu															
DESCRIPTION	<p>The nisaddcred command is used to create security credentials for NIS+ principals. NIS+ credentials serve two purposes. The first is to provide authentication information to various services; the second is to map the authentication service name into a NIS+ principal name.</p> <p>When the nisaddcred command is run, these credentials get created and stored in a table named cred.org_dir in the default NIS+ domain. If <i>domain_name</i> is specified, the entries are stored in the cred.org_dir of the specified domain. The specified domain must either be the one to which you belong, or one in which you are authenticated and authorized to create credentials, i.e., a subdomain. Note that the credentials of normal users must be stored in the same domain as their passwords.</p> <p>It is simpler to add credentials using nisclient(1M), because it obtains the required information itself. nispopulate(1M) is used for “bulk” updates and can also be used to add credentials for entries in the hosts and the passwd NIS+ tables.</p> <p>NIS+ principal names are used in specifying clients that have access rights to NIS+ objects. For more details, refer to the “Principal Names” subsection of the nis+(1) manual page. See nischmod(1), nischown(1), nis_objects(3N), and nis_groups(3N). Various other services can also implement access control based on these principal names.</p> <p>The cred.org_dir table is organized as follows :</p> <table border="1"> <thead> <tr> <th>cname</th> <th>auth_type</th> <th>auth_name</th> <th>public_data</th> <th>private_data</th> </tr> </thead> <tbody> <tr> <td>user1.foo.com.</td> <td>LOCAL</td> <td>2990</td> <td>10,102,44</td> <td></td> </tr> <tr> <td>user1.foo.com.</td> <td>DES</td> <td>unix.2990@foo.com</td> <td>098...819</td> <td>3b8...ab2</td> </tr> </tbody> </table> <p>The cname column contains a canonical representation of the NIS+ principal name. By convention, this name is the login name of a user, or the host name of a machine, followed by a dot (‘.’) followed by the fully qualified “home” domain of that principal. For users, the home domain is defined to be the domain where their DES credentials are kept. For hosts, their home domain is defined to be the domain name returned by the domainname(1M) command executed on that host.</p> <p>There are two types of <i>auth_type</i> entries in the cred.org_dir table. Those with authentication type LOCAL, and those with authentication type DES. <i>auth_type</i>, specified on the command line in upper or lower case, should be either <i>local</i> or <i>des</i>.</p> <p>Entries of type LOCAL are used by the NIS+ service to determine the correspondence between fully qualified NIS+ principal names and users identified by UIDs in the domain containing the cred.org_dir table.</p>	cname	auth_type	auth_name	public_data	private_data	user1.foo.com.	LOCAL	2990	10,102,44		user1.foo.com.	DES	unix.2990@foo.com	098...819	3b8...ab2
cname	auth_type	auth_name	public_data	private_data												
user1.foo.com.	LOCAL	2990	10,102,44													
user1.foo.com.	DES	unix.2990@foo.com	098...819	3b8...ab2												

This correspondence is required when associating requests made using the AUTH_SYS RPC authentication flavor (see `rpc_clnt_auth(3N)`) to a NIS+ principal name. It is also required for mapping a UID in one domain to its fully qualified NIS+ principal name whose home domain may be elsewhere. The principal's credentials for any authentication flavor may then be sought for within the `cred.org_dir` table in the principal's home domain (extracted from the principal name). The same NIS+ principal may have LOCAL credential entries in more than one domain. Only users, and not machines, have LOCAL credentials. In their home domain, users of NIS+ should have both types of credentials.

The `auth_name` associated with the LOCAL type entry is a UID that is valid for the principal in the domain containing the `cred.org_dir` table. This may differ from that in the principal's home domain. The public information stored in `public_data` for this type contains a list of GIDs for groups in which the user is a member. The GIDs also apply to the domain in which the table resides. There is no private data associated with this type. Neither a UID nor a principal name should appear more than once among the LOCAL entries in any one `cred.org_dir` table.

The DES `auth_type` is used for Secure RPC authentication (see `secure_rpc(3N)`).

The authentication name associated with the DES `auth_type` is a Secure RPC `netname`. A Secure RPC netname has the form `unix.id@domain.com`, where `domain` must be the same as the domain of the principal. For principals that are users the `id` must be the UID of the principal in the principal's home domain. For principals that are hosts, the `id` is the host's name. In Secure RPC, processes running under effective UID 0 (root) are identified with the host principal. Unlike LOCAL, there cannot be more than one DES credential entry for one NIS+ principal in the NIS+ namespace.

The public information in an entry of authentication type DES is the public key for the principal. The private information in this entry is the private key of the principal encrypted by the principal's network password.

User clients of NIS+ should have credentials of both types in their home domain. In addition, a principal must have a LOCAL entry in the `cred.org_dir` table of each domain from which the principal wishes to make authenticated requests. A client of NIS+ that makes a request from a domain in which it does not have a LOCAL entry will be unable to acquire DES credentials. A NIS+ service running at security level 2 or higher will consider such users unauthenticated and assign them the name `nobody` for determining access rights.

This command can only be run by those NIS+ principals who are authorized to add or delete the entries in the `cred` table.

If credentials are being added for the caller itself, `nisaddcred` automatically performs a keylogin for the caller.

You can list the `cred` entries for a particular principal with `nismatch(1)`.

OPTIONS

`-p principal` The name `principal` specifies the name of the principal as defined by the naming rules for that specific mechanism. For example, LOCAL credential names are supplied with this option by including a string specifying a UID. For DES credentials, the name should be a Secure RPC netname of the form `unix.id@domain.com`, as described earlier. If the `-p` option is not

specified, the *auth_name* field is constructed from the effective UID of the current process and the name of the local domain.

-P *nis_principal* Use the NIS+ principal name *nis_principal*. This option should be used when creating LOCAL or DES credentials for users whose home domain is different than the local machine's default domain.

Whenever the **-P** option is not specified, **nisaddcred** constructs a principal name for the entry as follows. When it is not creating an entry of type LOCAL, **nisaddcred** calls **nis_local_principal**, which looks for an existing LOCAL entry for the effective UID of the current process in the **cred.org_dir** table and uses the associated principal name for the new entry. When creating an entry of authentication type LOCAL, **nisaddcred** constructs a default NIS+ principal name by taking the login name of the effective UID for its own process, and appending to it a dot ('.') followed by the local machine's default domain. If the caller is a superuser, the machine name is used instead of the login name.

-l *login_password*

Use the *login_password* specified as the password to encrypt the secret key for the credential entry. This overrides the prompting for a password from the shell. This option is intended for administration scripts only. Prompting guarantees not only that no one can see your password on the command line using **ps(1)** but it also checks to make sure you have not made any mistakes. **NOTE:** *login_password* does not really HAVE to be the user's password but if it is, it simplifies logging in.

-r [*nis_principal*]

Remove all credentials associated with the principal *nis_principal* from the **cred.org_dir** table. This option can be used when removing a client or user from the system. If *nis_principal* is not specified the default is to remove credentials for the current *user*. If *domain_name* is not specified, the operation is executed in the default NIS+ domain.

EXIT CODES

This command returns **0** on success and **1** on failure.

EXAMPLES

The following examples illustrate how to add the LOCAL and DES credentials for some user, *user1*, with a UID of **2990**, who is an NIS+ user principal in the **some.domain.com**. NIS+ domain.

```
example% nisaddcred -p 2990 \  
-P user1.some.domain.com. local
```

Note that credentials are always added in the **cred.org_dir** table in the domain where **nisaddcred** is run, unless *domain_name* is specified as the last parameter on the command line. If credentials are being added from the domain server for its clients, then *domain_name* should be specified. The caller should have adequate permissions to create entries in the **cred.org_dir** table.

The system administrator can add a DES credential for the same user, using the following example:

```
example% nisaddcred -p unix.2990@some.domain.com \
-P user1.some.domain.com. des
```

Please note that DES credentials can be added only after the LOCAL credentials have been added.

Note that the secure RPC netname does not end with a dot ('.') while the NIS+ principal name (specified with the `-P` option) does. This command should be executed from a machine in the same domain as is the user.

The following example shows how to add a machine's DES credentials in the same domain.

```
example% nisaddcred -p unix.foo@some.domain.com \
-P foo.some.domain.com. des
```

Please note that no LOCAL credentials are needed in this case.

The following example illustrates how to add a NIS+ workstation's principal DES credential.

```
example% nisaddcred -p unix.host1@sub.some.domain.com \
-P newhost.sub.some.domain.com. des sub.some.domain.com.
```

This format is particularly useful if you are running this command from a server which is in a higher domain than `sub.some.domain.com`. Without the last option for domain name, `nisaddcred` would fail because it would attempt to use the default domain of `some.domain.com`.

The following example illustrates adding DES credentials without being prompted for the root login password.

```
example% nisaddcred -p unix.2990@some.domain.com \
-P user1.some.domain.com. -l login_password des
```

SEE ALSO

`chkey(1)`, `keylogin(1)`, `nis+(1)`, `nischmod(1)`, `nischown(1)`, `nismatch(1)`, `nistbladm(1)`, `ps(1)`, `domainname(M)`, `nisclient(1M)`, `nispopulate(1M)`, `nis_local_names(3N)`, `rpc_clnt_auth(3N)`, `secure_rpc(3N)`, `nis_objects(3N)`, `nis_groups(3N)`

NOTES

The `cred.org_dir` NIS+ table replaces the maps `publickey.byname` and `netid.byname` used in NIS (YP).

NAME	nisaddent – create NIS+ tables from corresponding /etc files or NIS maps
SYNOPSIS	<pre> /usr/lib/nis/nisaddent [-D defaults] [-Parv] [-t table] type [nisdomain] /usr/lib/nis/nisaddent [-D defaults] [-Paprvm] -f file [-t table] type [nisdomain] /usr/lib/nis/nisaddent [-D defaults] [-Parmv] [-t table] -y ypdomain [-Y map] type [nisdomain] /usr/lib/nis/nisaddent -d [-AMq] [-t table] type [nisdomain] </pre>
DESCRIPTION	<p>nisaddent creates entries in NIS+ tables from their corresponding /etc files and NIS maps. This operation is customized for each of the standard tables that are used in the administration of Solaris systems. The <i>type</i> argument specifies the type of the data being processed. Legal values for this type are one of aliases, bootparams, ethers, group, hosts, netid, netmasks, networks, passwd, protocols, publickey, rpc, services, shadow, or timezone for the standard tables, or key-value for a generic two-column (key, value) table. For a site specific table, which is not of key-value type, one can use nistbladm(1) to administer it.</p> <p>The NIS+ tables should have already been created by nistbladm(1), nissetup(1M), or nisserver(1M).</p> <p>It is easier to use nispopulate(1M) instead of nisaddent to populate the system tables.</p> <p>By default, nisaddent reads from the standard input and adds this data to the NIS+ table associated with the <i>type</i> specified on the command line. An alternate NIS+ table may be specified with the -t option. For type key-value, a table specification is required.</p> <p>Note that the <i>data</i> type can be different than the table name (-t). For example, the auto-mounter tables have key-value as the table type.</p> <p>Although, there is a <i>shadow</i> data type, there is no corresponding <i>shadow</i> table. Both the shadow and the passwd data is stored in the passwd table itself.</p> <p>Files may be processed using the -f option, and NIS version 2 (YP) maps may be processed using the -y option. The merge option is not available when reading data from standard input.</p> <p>When a <i>ypdomain</i> is specified, the nisaddent command takes its input from the dbm files for the appropriate NIS map (mail.aliases, bootparams, ethers.byaddr, group.byname, hosts.byaddr, netid.byname, netmasks.byaddr, networks.byname, passwd.byname, protocols.byname, publickey.byname, rpc.bynumber, services.byname, or timezone.byname). An alternate NIS map may be specified with the -Y option. For type key-value, a map specification is required. The map must be in the /var/yp/ypdomain directory on the local machine. Note that <i>ypdomain</i> is case sensitive. ypxfr(1M) can be used to get the NIS maps.</p> <p>If a <i>nisdomain</i> is specified, nisaddent operates on the NIS+ table in that NIS+ domain, otherwise the default domain is used.</p>

In terms of performance, loading up the tables is fastest when done through the dbm files (-y).

OPTIONS

- a Add the file or map to the NIS+ table without deleting any existing entries. This option is the default. Note that this mode only propagates additions and modifications, not deletions.
- p Process the password field when loading password information from a file. By default, the password field is ignored because it is usually not valid (the actual password appears in a shadow file).
- q Dump tables in "quick" mode. The default method for dumping tables processes each entry individually. For some tables (e.g., hosts), multiple entries must be combined into a single line, so extra requests to the server must be made. In "quick" mode, all of the entries for a table are retrieved in one call to the server, so the table can be dumped more quickly. However, for large tables, there is a chance that the process will run out of virtual memory and the table will not be dumped.
- r Replace the file or map in the existing NIS+ table by first deleting any existing entries, and then add the entries from the source (/etc files, or NIS+ maps). This option has the same effect as the -m option. The use of this option is *strongly* discouraged due to its adverse impact on performance, unless there are a large number of changes.
- m Merge the file or map with the NIS+ table. This is the most efficient way to bring an NIS+ table up to date with a file or NIS map when there are only a small number of changes. This option adds entries that are not already in the database, modifies entries that already exist (if changed), and deletes any entries that are not in the source. Use the -m option whenever the database is large and replicated, and the map being loaded differs only in a few entries. This option reduces the number of update messages that have to be sent to the replicas. Also see the -r option.
- d Dump the NIS+ table to the standard output in the appropriate format for the given *type*. For tables of type **key-value**, use **niscat(1)** instead. To dump the **cred** table, dump the **publickey** and the **netid** types.
- v Verbose.
- f *file* Specify that *file* should be used as the source of input (instead of the standard input).
- y *ypdomain* Use the **dbm** files for the appropriate NIS map, from the NIS domain *ypdomain*, as the source of input. The files are expected to be on the local machine in the **/var/yp/ypdomain** directory. If the machine is not an NIS server, use **ypxfr(1M)** to get a copy of the **dbm** files for the appropriate map.
- Y *map* Use the **dbm** files for *map* as the source of input.
- t *table* Specify that *table* should be the NIS+ table for this operation. This should be a

relative name as compared to your default domain or the *domainname* if it has been specified.

- P Follow concatenation path. This option specifies that lookups should follow the concatenation path of a table if the initial search is unsuccessful.
- A All data. This option specifies that the data within the table and all of the data in tables in the initial table's concatenation path be returned.
- M Master server only. This option specifies that lookups should be sent to the master server. This guarantees that the most up-to-date information is seen at the possible expense that the master server may be busy, or that it may be made busy by this operation.

-D defaults

This option specifies a different set of defaults to be used during this operation. The *defaults* string is a series of tokens separated by colons. These tokens represent the default values to be used for the generic object properties. All of the legal tokens are described below.

ttl=*time*

This token sets the default time to live for objects that are created by this command. The value *time* is specified in the format as defined by the **nischttl(1)** command. The default is 12 hours.

owner=*ownername*

This token specifies that the NIS+ principal *ownername* should own the created object. The default for this value is the principal who is executing the command.

group=*groupname*

This token specifies that the group *groupname* should be the group owner for the object that is created. The default is NULL.

access=*rights*

This token specifies the set of access rights that are to be granted for the given object. The value *rights* is specified in the format as defined by the **nischmod(1)** command. The default is `----rmdr---r---`.

EXAMPLES

This example adds the contents of `/etc/passwd` to the **passwd.org_dir** table.

```
example% cat /etc/passwd | nisaddent passwd
```

The next example adds the shadow information; note that the table type here is "shadow", not "passwd", even though the actual information is stored in the **passwd** table.

```
example% cat /etc/shadow | nisaddent shadow
```

This example replaces the **hosts.org_dir** table with the contents of `/etc/hosts` (in verbose mode).

```
example% nisaddent -rv -f /etc/hosts hosts
```

This example merges the **passwd** map from **myypdomain** with the **passwd.org_dir.nisdomain** table (in verbose mode). The example assumes that the **/var/yp/myypdomain** directory contains the **yppasswd** map.

```
example% nisaddent -mv -y myypdomain passwd nisdomain
```

This example merges the **auto.master** map from **myypdomain** with the **auto_master.org_dir** table.

```
example% nisaddent -m -y myypdomain -Y auto.master \
-t auto_master.org_dir key-value
```

This example dumps the **hosts.org_dir** table.

```
example% nisaddent -d hosts
```

ENVIRONMENT

NIS_DEFAULTS This variable contains a default string that will override the NIS+ standard defaults. If the **-D** switch is used, those values will then override both the **NIS_DEFAULTS** variable and the standard defaults. To avoid security accidents, the access rights in the **NIS_DEFAULTS** variable are ignored for the **passwd** table (but access rights specified with **-D** are used).

NIS_PATH If this variable is set, and neither the *nisdomain* nor the *table* are fully qualified, each directory specified in **NIS_PATH** will be searched until the table is found (see **nisdefaults(1)**).

EXIT CODES

- 0** Success.
- 1** Failure caused by an error other than parsing.
- 2** A parsing error occurred on an entry. A parsing error does not cause termination; the invalid entries are simply skipped.

SEE ALSO

niscat(1), **nischmod(1)**, **nisdefaults(1)**, **nistbladm(1)**, **nispopulate(1M)**, **nisserver(1M)**, **nissetup(1M)**, **ypxfr(1M)**, **hosts(4)**, **passwd(4)**, **shadow(4)**

NAME	nisclient – initialize NIS+ credentials for NIS+ principals
SYNOPSIS	<pre> /usr/lib/nis/nisclient -c [-x] [-o] [-v] [-l <network_password>] [-d <NIS+_domain>] client_name ... /usr/lib/nis/nisclient -i [-x] [-v] -h <NIS+_server_host> [-a <NIS+_server_addr>] [-d <NIS+_domain>] [-S 0/2] /usr/lib/nis/nisclient -u [-x] [-v] /usr/lib/nis/nisclient -r [-x] </pre>
DESCRIPTION	<p>The nisclient shell script can be used to:</p> <ul style="list-style-type: none"> • create NIS+ credentials for hosts and users • initialize NIS+ hosts and users • restore the network service environment <p>NIS+ credentials are used to provide authentication information of NIS+ clients to NIS+ service.</p> <p>Use the first synopsis (-c) to create individual NIS+ credentials for hosts or users. You must be logged in as a NIS+ principal in the domain for which you are creating the new credentials. You must also have write permission to the local "cred" table. The <i>client_name</i> argument accepts any valid host or user name in the NIS+ domain (for example, the <i>client_name</i> must exist in the hosts or passwd table). nisclient verifies each <i>client_name</i> against both the host and passwd tables, then adds the proper NIS+ credentials for hosts or users. Note that if you are creating NIS+ credentials outside of your local domain, the host or user must exist in the host or passwd tables in both the local and remote domains.</p> <p>By default, nisclient will not overwrite existing entries in the credential table for the hosts and users specified. To overwrite, use the -o option. After the credentials have been created, nisclient will print the command that must be executed on the client machine to initialize the host or the user. The -c option requires a network password for the client which is used to encrypt the secret key for the client. You can either specify it on the command line with the -l option or the script will prompt you for it. You can change this network password later with nispasswd(1) or chkey(1).</p> <p>nisclient -c is not intended to be used to create NIS+ credentials for all users and hosts which are defined in the passwd and hosts tables. To define credentials for all users and hosts, use nispopulate(1M).</p> <p>Use the second synopsis (-i) to initialize a NIS+ client machine. -i option can be used to convert machines to use NIS+ or to change the machine's domainname. You must be logged in as super-user on the machine that is to become a NIS+ client. Your administrator must have already created the NIS+ credential for this host by using nisclient -c or nispopulate -C. You will need the network password your administrator created. nisclient will prompt you for the network password to decrypt your secret key and then for this machine's root login password to generate a new set of secret/public keys. If the NIS+ credential was created by your administrator using nisclient -c, then you can</p>

simply use the initialization command that was printed by the **nisclient** script to initialize this host instead of typing it manually.

To initialize an unauthenticated NIS+ client machine, use the “-i” option with “-S 0”. With these options, the **nisclient -i** option will not ask for any passwords.

During the client initialization process, files that are being modified are backed up as <files>.no_nisplus. The files that are usually modified during a client initialization are: /etc/defaultdomain, /etc/nsswitch.conf, /etc/inet/hosts, and, if it exists, /var/nis/NIS_COLD_START. Note that a file will not be saved if a backup file already exists.

The -i option does not set up an NIS+ client to resolve hostnames using DNS. Please refer to the DNS documentation for information on setting up DNS. (See **resolv.conf(4)**).

It is not necessary to initialize either NIS+ root master servers or machines that were installed as NIS+ clients using **suninstall(1M)**.

Use the third synopsis (-u) to initialize a NIS+ user. You must be logged in as the user on a NIS+ client machine in the domain where your NIS+ credentials have been created. Your administrator should have already created the NIS+ credential for your username using **nisclient -c** or **nispopulate(1M)**. You will need the network password your administrator used to create the NIS+ credential for your username. **nisclient** will prompt you for this network password to decrypt your secret key and then for your login password to generate a new set of secret/public keys.

Use the fourth synopsis (-r) to restore the network service environment to whatever you were using before **nisclient -i** was executed. You must be logged in as super-user on the machine that is to be restored. The restore will only work if the machine was initialized with **nisclient -i** because it uses the backup files created by the -i option.

Reboot the machine after initializing a machine or restoring the network service.

OPTIONS

- a <NIS+_server_addr> Specifies the IP address for the NIS+ server. This option is used *only* with the -i option.
- c Adds DES credentials for NIS+ principals.
- d <NIS+_domain> Specifies the NIS+ domain where the credential should be created when used in conjunction with the -c option. It specifies the name for the new NIS+ domain when used in conjunction with the -i option. The default is your current domainname.
- h <NIS+_server_host> Specifies the NIS+ server’s hostname. This option is used *only* with the -i option.
- i Initializes an NIS+ client machine.
- l <network_password> Specifies the network password for the clients. This option is used *only* with the -c option. If this option is not specified, the script will prompt you for the network password.
- o Overwrite existing credential entries. The default is not to overwrite. This is used *only* with the -c option.

-r restores the network service environment.

-S 0/2 Specifies the authentication level for the NIS+ client. Level **0** is for unauthenticated clients and level **2** is for authenticated (**DES**) clients. The default is to set up with level **2** authentication. This is used *only* with the **-i** option. **nisclient** always uses level **2** authentication (**DES**) for both **-c** and **-u** options. There is no need to run **nisclient** with **-u** and **-c** for level **0** authentication.

-u Initializes an NIS+ user.

-v Runs the script in verbose mode.

-x turns the "echo" mode on. The script just prints the commands that it would have executed. Note that the commands are not actually executed. The default is off.

EXAMPLES

To add the **DES** credential for host *sunws* and user *fred* in the local domain:

```
example% /usr/lib/nis/nisclient -c sunws fred
```

To add the **DES** credential for host *sunws* and user *fred* in domain *xyz.sun.com.*:

```
example% /usr/lib/nis/nisclient -c -d xyz.sun.com. sunws fred
```

To initialize host *sunws* as an NIS+ client in domain *xyz.sun.com.* where *nisplus_server* is a server for the domain *xyz.sun.com.*:

```
example# /usr/lib/nis/nisclient -i -h nisplus_server -d xyz.sun.com.
```

The script will prompt you for the IP address of *nisplus_server* if the server is not found in the **/etc/hosts** file. The **-d** option is needed only if your current domain name is different from the new domain name.

To initialize host *sunws* as an unauthenticated NIS+ client in domain *xyz.sun.com.* where *nisplus_server* is a server for the domain *xyz.sun.com.*:

```
example# /usr/lib/nis/nisclient -i -S 0 \  
-h nisplus_server -d xyz.sun.com. -a 129.140.44.1
```

To initialize user *fred* as an NIS+ principal, log in as user *fred* on an NIS+ client machine.

```
example% /usr/lib/nis/nisclient -u
```

FILES

/var/nis/NIS_COLD_START

This file contains a list of servers, their transport addresses, and their Secure RPC public keys that serve the machines default domain.

/etc/defaultdomain the system default domainname

/etc/nsswitch.conf configuration file for the name-service switch

/etc/inet/hosts local host name database

SEE ALSO

**chkey(1), keylogin(1), nis+(1), nispasswd(1), keyserv(1M), nisaddcred(1M),
nisinit(1M), nispopulate(1M), suninstall(1M), nsswitch.conf(4), resolv.conf(4)**

NAME	nisinit – NIS+ client and server initialization utility
SYNOPSIS	<pre>nisinit -r nisinit -p Y D N <i>parent_domain host</i> ... nisinit -c -H <i>host</i> -B -C <i>coldstart</i></pre>
AVAILABILITY	SUNWnisu
DESCRIPTION	nisinit initializes a machine to be a NIS+ client or an NIS+ root master server. It may be easier to use nisclient (1M) or nissserver (1M) to accomplish this same task.
OPTIONS	<p>-r Initialize the machine to be a NIS+ root server. This option creates the file <code>/var/nis/data/root.object</code> and initialize it to contain information about this machine. It uses the sysinfo(2) system call to retrieve the name of the default domain.</p> <p>To initialize the machine as an NIS+ root server, it is advisable to use the “-r” option of nissserver(1M), instead of using “nisinit -r”.</p> <p>-p Y D N <i>parent_domain host</i> ...</p> <p>This option is used on a root server to initialize a <code>/var/nis/data/parent.object</code> to make this domain a part of the namespace above it. Only root servers can have parent objects. A parent object describes the namespace “above” the NIS+ root. If this is an isolated domain, this option should not be used. The argument to this option tells the command what type of name server is serving the domain above the NIS+ domain. When clients attempt to resolve a name that is outside of the NIS+ namespace, this object is returned with the error NIS_FOREIGNNS indicating that a name space boundary has been reached. It is up to the client to continue the name resolution process.</p> <p>The parameter <i>parent_domain</i> is the name of the parent domain in a syntax that is native to that type of domain. The list of host names that follow the domain parameter are the names of hosts that serve the parent domain. If there is more than one server for a parent domain, the first host specified should be the master server for that domain.</p> <p>Y Specifies that the parent directory is a NIS version 2 domain.</p> <p>D Specifies that the parent directory is a DNS domain.</p> <p>N Specifies that the parent directory is another NIS+ domain. This option is useful for connecting a pre-existing NIS+ subtree into the global namespace.</p> <p>Note that in the current implementation, the NIS+ clients do not take advantage of the -p feature. Also, since the parent object is currently not replicated on root replica servers, it is recommended that this option not be used.</p>

- c** Initializes the machine to be a NIS+ client. There are three initialization options available: initialize by coldstart, initialize by hostname, and initialize by broadcast. The most secure mechanism is to initialize from a trusted coldstart file. The second option is to initialize using a hostname that you specify as a trusted host. The third method is to initialize by broadcast and it is the least secure method.
- C coldstart**
Causes the file *coldstart* to be used as a prototype coldstart file when initializing a NIS+ client. This coldstart file can be copied from a machine that is already a client of the NIS+ namespace. For maximum security, an administrator can encrypt and encode (with **uuencode(1C)**) the coldstart file and mail it to an administrator bringing up a new machine. The new administrator would then decode (with **uudecode**), decrypt, and then use this file with the **nisinit** command to initialize the machine as an NIS+ client. If the coldstart file is from another client in the same domain, the **nisinit** command may be safely skipped and the file copied into the **/var/nis** directory as **/var/nis/NIS_COLD_START**.
- H hostname**
Specifies that the host *hostname* should be contacted as a trusted NIS+ server. The **nisinit** command will iterate over each transport in the **NETPATH** environment variable and attempt to contact **rpcbind(1M)** on that machine. This hostname *must* be reachable from the client without the name service running. For IP networks this means that there must be an entry in **/etc/hosts** for this host when **nisinit** is invoked.
- B** Specifies that the **nisinit** command should use an IP broadcast to locate a NIS+ server on the local subnet. Any machine that is running the NIS+ service may answer. No guarantees are made that the server that answers is a server of the organization's namespace. If this option is used, it is advisable to check with your system administrator that the server and domain served are valid. The binding information can be dumped to the standard output using the **nisshowcache(1M)** command.

Note that **nisinit -c** will just enable navigation of the NIS+ name space from this client. To make NIS+ your name service, modify the file **/etc/nsswitch.conf** to reflect that. See **nsswitch.conf(4)** for more details.

RETURN VALUES

nisinit returns **0** on success and **1** on failure.

EXAMPLES

This example initializes the machine as an NIS+ client using the host *freddy* as a trusted server.

```
example# nisinit -cH freddy
```

This example sets up a client using a trusted coldstart file.

```
example# nisinit -cC /tmp/colddata
```

This example sets up a client using an IP broadcast.

```
example# nisinit -cB
```

This example sets up a root server.

```
example# nisinit -r
```

ENVIRONMENT

NETPATH This environment variable may be set to the transports to try when contacting the NIS+ server (see **netconfig(4)**). The client library will only attempt to contact the server using connection oriented transports.

FILES

/var/nis/NIS_COLD_START

This file contains a list of servers, their transport addresses, and their Secure RPC public keys that serve the machine's default domain.

/var/nis/data/root.object

This file describes the root object of the NIS+ namespace. It is a standard XDR-encoded NIS+ directory object that can be modified by authorized clients using the **nis_modify()** interface.

/var/nis/data/parent.object

This file describes the namespace that is logically above the NIS+ namespace. The most common type of parent object is a DNS object. This object contains contact information for a server of that domain.

/etc/hosts

Internet host table.

SEE ALSO

nis+(1), **uuencode(1C)**, **nisclient(1M)**, **nissserver(1M)**, **nisshowcache(1M)**, **sysinfo(2)**, **hosts(4)**, **netconfig(4)**, **nisfiles(4)**

NAME	nislog – display the contents of the NIS+ transaction log
SYNOPSIS	/usr/sbin/nislog [-h num -t num] [-v] [<i>directory ...</i>]
AVAILABILITY	SUNWnisu
DESCRIPTION	<p>nislog displays the contents of the NIS+ server transaction log on the standard output. This command can be used to track changes in the namespace. The /var/nis/trans.log file contains the transaction log maintained by the NIS+ server. When updates occur, they are logged to this file and then propagated to replicas as log transactions. When the log is checkpointed, updates that have been propagated to the replicas are removed.</p> <p>The nislog command can only be run on an NIS+ server by superuser. It displays the log entries for that server only.</p> <p>If <i>directory</i> is not specified, the entire log is searched. Otherwise, only those logs entries that correspond to the specified directories are displayed.</p>
OPTIONS	<p>-h num Display <i>num</i> transactions from the “head” of the log. If the numeric parameter is 0, only the log header is displayed.</p> <p>-t num Display <i>num</i> transactions from the “tail” of the log. If the numeric parameter is 0, only the log header is displayed.</p> <p>-v Verbose mode.</p>
FILES	/var/nis/trans.log transaction log
SEE ALSO	nis+(1) , rpc.nisd(1M) , nisfiles(4)

NAME	nisping – send ping to NIS+ servers
SYNOPSIS	<code>/usr/lib/nis/nisping [-uf] [-H <i>hostname</i>] [-r <i>directory</i>]</code> <code>/usr/lib/nis/nisping -C [-a] [-H <i>hostname</i>] [<i>directory</i>]</code>
DESCRIPTION	<p>In the first SYNOPSIS line, the nisping command sends a “ping” to all replicas of a NIS+ directory. Once a replica receives a ping, it will check with the master server for the directory to get updates. Prior to pinging the replicas, this command attempts to determine the last update “seen” by a replica and the last update logged by the master. If these two timestamps are the same, the ping is not sent. The -f (force) option will override this feature.</p> <p>Under normal circumstances, NIS+ replica servers get the new information from the master NIS+ server within a short time. Therefore, there should not be any need to use nisping.</p> <p>In the second SYNOPSIS line, the nisping -C command sends a checkpoint request to the servers. If no <i>directory</i> is specified, the home domain, as returned by nisdefaults(1), is checkpointed. If all directories, served by a given server, have to be checkpointed, then use the -a option.</p> <p>On receiving a checkpoint request, the servers would commit all the updates for the given <i>directory</i> from the table log files to the database files. This command, if sent to the master server, will also send updates to the replicas if they are out of date. This option is needed because the database log files for NIS+ are not automatically checkpointed. nisping should be used at frequent intervals (such as once a day) to checkpoint the NIS+ database log files. This command can be added to the crontab(1) file. If the database log files are not checkpointed, their sizes will continue to grow.</p>
OPTIONS	<p>-a Checkpoint all directories on the server.</p> <p>-C Send a request to checkpoint, rather than a ping, to each server. The servers schedule to commit all the transactions to stable storage.</p> <p>-H <i>hostname</i> Only the host <i>hostname</i> is sent the ping, checked for an update time, or checkpointed.</p> <p>-f Force a ping, even though the timestamps indicate there is no reason to do so. This option is useful for debugging.</p> <p>-r This option can be used to update or get status about the root object from the root servers, especially when new root replicas are added or deleted from the list.</p> <p>If used without -u option, -r will send a ping request to the servers serving the root domain. When the replicas receive a ping, they will update their root object if needed.</p> <p>The -r option can be used with all other options except with the -C option; the root object need not be checkpointed.</p>

	-u	Display the time of the last update; no servers are sent a ping.
RETURN VALUES	-1	No servers were contacted, or the server specified by the -H switch could not be contacted.
	0	Success.
	1	Some, but not all, servers were successfully contacted.
EXAMPLES	This example pings all replicas of the default domain: example% nisping	
	Note that this example will not ping the the org_dir and group_dir subdirectories within this domain.	
	This example pings the server <i>example</i> which is a replica of the <i>org_dir.foo.com</i> . directory: example% nisping -H example org_dir.foo.com.	
	This example checkpoints all servers of the <i>org_dir.bar.com</i> . directory. example% nisping -C org_dir.bar.com.	
ENVIRONMENT	NIS_PATH	If this variable is set, and the NIS+ directory name is not fully qualified, each directory specified will be searched until the directory is found.
SEE ALSO	crontab(1) , nisdefaults(1) , nislog(1M) , nisfiles(4)	
NOTES	If the server specified by the -H option does not serve the directory, then no ping is sent.	

NAME	nispopulate – populate the NIS+ tables in a NIS+ domain.
SYNOPSIS	<pre> /usr/lib/nis/nispopulate -Y [-x] [-f] [[-u] [-v] [-S 0 2] [-I <network_passwd>] [-d <NIS+_domain>] -h <NIS_server_host> [-a <NIS_server_addr>] -y <NIS_domain> [table] ... /usr/lib/nis/nispopulate -F [-x] [[-f] [-u] [-v] [-S 0 2] [-d <NIS+_domain>] [-I <network_passwd>] [-p <directory_path>] [table] ... /usr/lib/nis/nispopulate -C [-x] [-f] [-v] [-d <NIS+_domain>] [-I <network_passwd>] [hosts / passwd]] </pre>
DESCRIPTION	<p>The nispopulate shell script can be used to populate NIS+ tables in a specified domain from their corresponding files or NIS maps. nispopulate assumes that the tables have been created either through nisserv(1M) or nissetup(1M).</p> <p>The table argument accepts standard names that are used in the administration of Solaris systems and non-standard <i>key-value</i> type tables. See nisaddent(1M) for more information on <i>key-value</i> type tables. If the table argument is not specified, nispopulate will automatically populate each of the standard tables. These standard (default) tables are: auto_master, auto_home, ethers, group, hosts, networks, passwd, protocols, services, rpc, netmasks, bootparams, netgroup, aliases and shadow. Note that the shadow table is only used when populating from files. The non-standard tables that nispopulate accepts are those of <i>key-value</i> type. These tables must first be created manually with the nistbladm(1) command.</p> <p>Use the first synopsis (-Y) to populate NIS+ tables from NIS maps. nispopulate uses ypxfr(1M) to transfer the NIS maps from the NIS servers to the /var/yp/<NIS_domain> directory on the local machine. Then, it uses these files as the input source. Note that <NIS_domain> is case sensitive. Make sure there is enough disk space for that directory.</p> <p>Use the second synopsis (-F) to populate NIS+ tables from local files. nispopulate will use those files that match the table name as input sources in the current working directory or in the specified directory.</p> <p>Note that when populating the hosts and passwd tables, nispopulate will automatically create the NIS+ credentials for all users and hosts which are defined in the hosts and passwd tables, respectively. A network passwd is required to create these credentials. This network password is used to encrypt the secret key for the new users and hosts. This password can be specified using the -I option or it will use the default password, "nisplus". nispopulate will not overwrite any existing credential entries in the credential table. Use nisclient(1M) to overwrite the entries in the cred table. It creates both LOCAL and DES credentials for users, and only DES credentials for hosts. To disable automatic credential creation, specify the "-S 0" option.</p>

The third synopsis (-C) is used to populate NIS+ credential table with level 2 authentication (**DES**) from the passwd and hosts tables of the specified domain. The valid table arguments for this operation are passwd and hosts. If this argument is not specified then it will use both passwd and hosts as the input source.

If **nispopulate** was earlier used with "-S 0" option, then no credentials were added for the hosts or the users. If later the site decides to add credentials for all users and hosts, then this (-C) option can be used to add credentials.

OPTIONS

- a <NIS_server_addr> specifies the IP address for the NIS server. This option is ONLY used with the -Y option.
- C populate the NIS+ credential table from passwd and hosts tables using **DES** authentication (security level 2).
- d <NIS+_domain.> specifies the NIS+ domain. The default is the local domain.
- F populates NIS+ tables from files.
- f forces the script to populate the NIS+ tables without prompting for confirmation.
- h <NIS_server_host> specifies the NIS server hostname from where the NIS maps are copied from. This is ONLY used with the -Y option. This host must be already exist in either the NIS+ **hosts** table or **/etc/hosts** file. If the hostname is not defined, the script will prompt you for its IP address, or you can use the -a option to specify the address manually.
- l <network_passwd> specifies the network password for populating the NIS+ credential table. This is ONLY used when you are populating the **hosts** and **passwd** tables. The default passwd is "nisplus".
- n does not overwrite local NIS maps in **/var/yp/<NISdomain>** directory if they already exist. The default is to overwrite the existing NIS maps in the local **/var/yp/<NISdomain>** directory. This is ONLY used with the -Y option.
- p <directory_path> specifies the directory where the files are stored. This is ONLY used with the -F option. The default is the current working directory.
- S 0/2 specifies the authentication level for the NIS+ clients. Level 0 is for unauthenticated clients and no credentials will be created for users and hosts in the specified domain. Level 2 is for authenticated (**DES**) clients and DES credentials will be created for users and hosts in the specified domain. The default is to set up with level 2 authentication (**DES**).
There is no need to run nispopulate with -C for level 0 authentication.

- u** updates the NIS+ tables (ie., adds, deletes, modifies) from either files or NIS maps. This option should be used to bring an NIS+ table up to date when there are only a small number of changes. The default is to add to the NIS+ tables without deleting any existing entries. Also, see the **-n** option for updating NIS+ tables from existing maps in the **/var/yp** directory.
- v** runs the script in verbose mode.
- x** turns the "echo" mode on. The script just prints the commands that it would have executed. Note that the commands are not actually executed. The default is off.
- Y** populate the NIS+ tables from NIS maps.
- y <NIS_domain>** specifies the NIS domain to copy the NIS maps from. This is ONLY used with the **-Y** option. The default domainname is the same as the local domainname.

ENVIRONMENT

TMPDIR

nispopulate normally creates temporary files in the directory **/tmp**. You may specify another directory by setting the environment variable **TMPDIR** to your chosen directory. If **TMPDIR** is not a valid directory, then **nispopulate** will use **/tmp**.

EXAMPLES

To populate all the NIS+ standard tables in the domain *xyz.sun.com*. from NIS maps of the *yp.sun.COM* domain as input source where host *yp_host* is a YP server of *yp.sun.COM*:

```
nis_server# /usr/lib/nis/nispopulate -Y -y yp.sun.COM -h yp_host -d xyz.sun.com.
```

To update all of the NIS+ standard tables from the same NIS domain and hosts shown above:

```
nis_server# /usr/lib/nis/nispopulate -Y -u -y yp.sun.COM -h yp_host \  
-d xyz.sun.com.
```

To populate the **hosts** table in domain *xyz.sun.com*. from the hosts file in the **/var/nis/files** directory and using "somepasswd" as the network password for key encryption:

```
nis_server# /usr/lib/nis/nispopulate -F -p /var/nis/files -l somepasswd hosts
```

To populate the passwd table in domain *xyz.sun.com*. from the passwd file in the **/var/nis/files** directory without automatically creating the NIS+ credentials:

```
nis_server# /usr/lib/nis/nispopulate -F -p /var/nis/files  
-d xys.sun.com. -S 0 passwd
```

To populate the credential table in domain *xyz.sun.com*. for all users defined in the passwd table.

```
nis_server# /usr/lib/nis/nispopulate -C -d xys.sun.com. passwd
```

To create and populate a non-standard key-value type NIS+ table, "private", from the file **/var/nis/files/private**: (**nispopulate** assumes that the private.org_dirkey-value type table has already been created).

```
nis_server# /usr/bin/nistbladm -D access=og=rmcd,nw=r \  
-c private key=S,nogw= value=,nogw= private.org.dir  
nis_server# /usr/lib/nis/nispopulate -F -p /var/nis/files private
```

FILES

/etc/hosts	local host name database
/var/yp	NIS(YP) domain directory
/var/nis	NIS+ domain directory
/tmp	

SEE ALSO **nis+(1), nistbladm(1), nisaddcred(1M), nisaddent(1M), nisclient(1M), nisserver(1M), nissetup(1M), rpc.nisd(1M), ypxfr(1M)**

NAME	nisserver – set up NIS+ servers.
SYNOPSIS	<pre> /usr/lib/nis/nisserver -r [-x] [-f] [-v][-Y] [-d <NIS+_domain>] [-g <NIS+_groupname>] [-l <network_passwd>] /usr/lib/nis/nisserver -M [-x] [-f] [-v] [-Y] [-d <NIS+_domain>] [-g <NIS+_groupname>] [-h <NIS+_server_host>] /usr/lib/nis/nisserver -R [-x] [-f] [-v] [-Y] [-d <NIS+_domain>] [-h <NIS+_server_host>] </pre>
DESCRIPTION	<p>The nisserver shell script can be used to set up a root master, non-root master, and replica NIS+ servers with level 2 security (DES).</p> <p>When setting up a new domain, this script creates the NIS+ directories (including groups_dir and org_dir) and system table objects for the domain specified. It does not populate the tables. You will need to use nispopulate(1M) to populate the tables.</p> <p>Use the first synopsis (-r) to set up a root master server. You must be logged in as super-user on the server machine.</p> <p>Use the second synopsis (-M) to set up a non-root master server for the specified domain. You must be logged in as an NIS+ principal on a NIS+ machine and have write permission to the parent directory of the domain that you are setting up. The new non-root master server machine must already be an NIS+ client (see nisclient(1M)) and have the rpc.nisd daemon running (see rpc.nisd(1M)).</p> <p>Use the third synopsis (-R) to set up a replica server for both root and non-root domains. You must be logged in as an NIS+ principal on an NIS+ machine and have write permission to the parent directory of the domain that you are replicating. The new non-root replica server machine must already be an NIS+ client (see nisclient(1M)) and have the rpc.nisd daemon running (see rpc.nisd(1M)).</p>
OPTIONS	<p>-d <NIS+_domain> specifies the name for the NIS+ domain. The default is your local domain.</p> <p>-f forces the NIS+ server setup without prompting for confirmation.</p> <p>-g <NIS+_groupname> specifies the NIS+ group name for the new domain. This option is not valid with -R option. The default group is admin.<domain>.</p> <p>-h <NIS+_server_host> specifies the hostname for the NIS+ server. It must be a valid host in the local domain. Use a fully qualified hostname (for example, hostx.xyz.sun.com.) to specify a host outside of your local domain. This option is ONLY used for setting up non-root master or replica servers. The default for non-root master server setup is to use the same list of servers as the parent domain. The default for replica server setup is the local hostname.</p>

- l** *<network_password>* specifies the network password with which to create the credentials for the root master server. This option is ONLY used for master root server setup (**-r** option). If this option is not specified, the script will prompt you for the login password.
- M** sets up the specified host as a master server. Make sure that **rpc.nisd(1M)** is running on the new master server before this command is executed.
- R** sets up the specified host as a replica server. Make sure that **rpc.nisd(1M)** is running on the new replica server.
- r** sets up the server as a root master server. Use the **-R** option to set up a root replica server.
- v** runs the script in verbose mode.
- x** turns the "echo" mode on. The script just prints the commands that it would have executed. Note that the commands are not actually executed. The default is off.
- Y** sets up an NIS+ server with NIS-compatibility mode. The default is to set up the server without NIS-compatibility mode.

EXAMPLES

To set up a root master server for domain *sun.com*. :

```
root_server# /usr/lib/nis/nisserver -r -d sun.com.
```

For the following examples make sure that the new servers are NIS+ clients and **rpc.nisd** is running on these hosts before executing **nisserver**.

To set up a replica server for domain *sun.com*. on host *sunreplica* :

```
root_server# /usr/lib/nis/nisserver -R -d sun.com. -h sunrep
```

To set up a non-root master server for domain *xyz.sun.com*. on host *sunxyz* with the NIS+ groupname as *admin-mgr.xyz.sun.com*. :

```
root_server# /usr/lib/nis/nisserver -M -d xyz.sun.com. -h sunxyz \  
-g admin-mgr.xyz.sun.com.
```

To set up a non-root replica server for domain *xyz.sun.com*. on host *sunabc*:

```
sunxyz# /usr/lib/nis/nisserver -R -d xyz.sun.com. -h sunabc
```

SEE ALSO

nis+(1), **nisgrpadm(1)**, **nismkdir(1)**, **nisaddcred(1M)**, **nisclient(1M)**, **nisinit(1M)**, **nismkdir(1)**, **nispopulate(1M)**, **nissetup(1M)**, **rpc.nisd(1M)**

NAME	<code>nissetup</code> – initialize a NIS+ domain
SYNOPSIS	<code>/usr/lib/nis/nissetup</code> [<code>-Y</code>] [<i>domain</i>]
DESCRIPTION	<p>nissetup is a shell script that sets up a NIS+ domain to service clients that wish to store system administration information in a domain named <i>domain</i>. This domain should already exist prior to executing this command (see nismkdir(1) and nisinit(1M)).</p> <p>A NIS+ domain consists of a NIS+ directory and its subdirectories: org_dir and groups_dir. org_dir stores system administration information and groups_dir stores information for group access control.</p> <p>nissetup creates the subdirectories org_dir and groups_dir in <i>domain</i>. Both subdirectories will be replicated on the same servers as the parent domain. After the subdirectories are created, nissetup creates the default tables that NIS+ serves. These are auto_master, auto_home, bootparams, cred, ethers, group, hosts, mail_aliases, net-masks, networks, passwd, protocols, rpc, services, and timezone. The nissetup script uses the nistbladm(1) command to create these tables. The script can be easily customized to add site specific tables that should be created at setup time.</p> <p>This command is normally executed just once per domain.</p>
OPTIONS	<p><code>-Y</code> Specify that the domain will be served as both a NIS+ domain as well as an NIS domain using the backward compatibility flag. This will set up the domain to be less secure by making all the system tables readable by unauthenticated clients as well.</p>
SEE ALSO	nis+ (1), nismkdir (1), nistbladm (1), nisaddent (1M), nisinit (1M) nisserver (1M),
NOTES	<p>While this command creates the default tables, it does not initialize them with data. This is accomplished with the nisaddent(1M) command.</p> <p>It is easier to use the nisserver(1M) script to create subdirectories and the default tables.</p>

NAME	nisshowcache – NIS+ utility to print out the contents of the shared cache file
SYNOPSIS	/usr/lib/nis/nisshowcache [-v]
DESCRIPTION	nisshowcache prints out the contents of the per-machine NIS+ directory cache that is shared by all processes accessing NIS+ on the machine. By default, nisshowcache only prints out the directory names in the cache along with the cache header. The shared cache is maintained by nis_cachemgr (1M).
OPTIONS	-v Verbose mode. Print out the contents of each directory object, including information on the server name and its universal addresses.
FILES	/var/nis/NIS_SHARED_DIRCACHE
SEE ALSO	nis_cachemgr (1M), syslogd (1M), nisfiles (4)
DIAGNOSTICS	Error messages are sent to the syslogd (1M) daemon.

NAME	nisstat – report NIS+ server statistics
SYNOPSIS	<code>/usr/lib/nis/nisstat [-H <i>host</i>] [<i>directory</i>]</code>
DESCRIPTION	<p>The nisstat command queries a NIS+ server for various statistics about its operations. These statistics may vary between implementations and from release to release. Not all statistics are available from all servers. Requesting a statistic from a server that does not support that statistic is never fatal, it simply returns “unknown statistic”.</p> <p>By default, statistics are fetched from the server(s) of the NIS+ directory for the default domain. If <i>directory</i> is specified, servers for that directory are queried.</p> <p>Supported statistics for this release are as follows:</p> <p><i>root server</i> This reports whether the server is a root server.</p> <p><i>NIS compat mode</i> This reports whether the server is running in NIS compat mode.</p> <p><i>DNS forwarding in NIS mode</i> This reports whether the server in NIS compat mode will forward host lookup calls to DNS.</p> <p><i>security level</i> This reports the security level of this server.</p> <p><i>serves directories</i> This lists the directories served by this server.</p> <p><i>Operations</i> This statistic returns results in the form: OP=opname:C=calls:E=errors:T=micros Where <i>opname</i> is replaced by the RPC procedure name or operation, <i>calls</i> is the number of calls to this procedure that have been made since the server started running. <i>errors</i> is the number of errors that have occurred while processing a call, and <i>micros</i> is the average time in microseconds to complete the last 16 calls.</p> <p><i>Directory Cache</i> This statistic reports the number of calls to the internal directory object cache, the number of hits on that cache, the number of misses, and the hit rate percentage.</p> <p><i>Group Cache</i> This statistic reports the number of calls to the internal NIS+ group object cache, the number of hits on that cache, the number of misses, and the hit rate percentage.</p> <p><i>Static Storage</i> This statistic reports the number of bytes the server has allocated for its static storage buffers.</p> <p><i>Dynamic Storage</i> This statistic reports the amount of heap the server process is currently using.</p> <p><i>Uptime</i> This statistic reports the time since the service has been running.</p>
OPTIONS	<p>-H <i>host</i> Normally all servers for the directory are queried. With this option, only the machine named <i>host</i> is queried. If the named machine does not serve the directory, no statistics are returned.</p>

ENVIRONMENT**NIS_PATH**

If this variable is set, and the NIS+ directory name is not fully qualified, each directory specified will be searched until the directory is found (see **nisdefaults(1)**).

SEE ALSO**nisdefaults(1)**

NAME	nisupdkeys – update the public keys in a NIS+ directory object
SYNOPSIS	<pre>/usr/lib/nis/nisupdkeys [-a -C] [-H host] [directory] /usr/lib/nis/nisupdkeys -s [-a -C] -H host</pre>
DESCRIPTION	<p>This command updates the public keys in an NIS+ directory object. When the public key for a NIS+ server is changed, the new key must be propagated to all directory objects that reference that server.</p> <p>nisupdkeys reads a directory object and attempts to get the public key for each server of that directory. These keys are placed in the directory object and the object is then modified to reflect the new keys.</p> <p>If <i>directory</i> is present, the directory object for that directory is updated. Otherwise the directory object for the default domain is updated.</p> <p>On the other hand, nisupdkeys -s gets a list of all the directories served by <i>host</i> and updates those directory objects. This assumes that the caller has adequate permission to change all the associated directory objects. The list of directories being served by a given server can also be obtained by nisstat(1M).</p> <p>Before you do this operation, make sure that the new address/public key has been propagated to all replicas.</p>
OPTIONS	<p>-a Update the universal addresses of the NIS+ servers in the directory object. Currently, this only works for the TCP/IP family of transports. This option should be used when the IP address of the server is changed. The server's new address is resolved using gethostbyname(3N) on this machine. The /etc/nsswitch.conf file must point to the correct source for the <i>hosts</i> entry for this resolution to work.</p> <p>-C Specify to clear rather than set the public key. Communication with a server that has no public key does not require the use of secure RPC.</p> <p>-H host Limit key changes only to the server named <i>host</i>. If the hostname is not a fully qualified NIS+ name, then it is assumed to be a host in the default domain. If the named host does not serve the directory, no action is taken.</p> <p>-s Update all the NIS+ directory objects served by the specified server. This assumes that the caller has adequate access rights to change all the associated directory objects. If the NIS+ principal making this call does not have adequate permissions to update the directory objects, those particular updates will fail and the caller will be notified. If the rpc.nisd on <i>host</i> cannot return the list of servers it serves, the command will print an error message. The caller would then have to invoke nisupdkeys multiple times (as in the first synopsis), once per NIS+ directory that it serves.</p>
EXAMPLES	<p>The following example updates the keys for servers of the <i>foo.bar</i>. domain.</p> <pre>example% nisupdkeys foo.bar.</pre>

This example updates the key for host *fred* which serves the *foo.bar*. domain.

```
example% nisupdkeys -H fred foo.bar.
```

This example clears the public key for host *wilma* in the *foo.bar*. directory.

```
example% nisupdkeys -CH wilma foo.bar.
```

This example updates the public key in all directory objects that are served by the host *wilma*.

```
example% nisupdkeys -s -H wilma
```

SEE ALSO

chkey(1), **niscat(1)**, **nisaddcred(1M)**, **gethostbyname(3N)**, **nis_objects(3N)**

NOTES

The user executing this command must have modify access to the directory object for it to succeed. The existing directory object can be displayed with the **niscat(1)** command using the **-o** option.

This command does not update the directory objects stored in the **NIS_COLD_START** file on the NIS+ clients.

If a server is also the root master server, then **nisupdkeys -s** cannot be used to update the root directory.

NAME	nlsadmin – network listener service administration
SYNOPSIS	<pre> /usr/sbin/nlsadmin -x /usr/sbin/nlsadmin [options] net_spec /usr/sbin/nlsadmin [options] -N port_monitor_tag /usr/sbin/nlsadmin -V /usr/sbin/nlsadmin -c cmd -o streamname [-p modules] [-A address -D] [-R prognum: versnum] </pre>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>nlsadmin is the administrative command for the network listener process(es) on a machine. Each network has at least one instance of the network listener process associated with it; each instance (and thus, each network) is configured separately. The listener process “listens” to the network for service requests, accepts requests when they arrive, and invokes servers in response to those service requests. The network listener process may be used with any network (more precisely, with any connection-oriented transport provider) that conforms to the transport provider specification.</p> <p>nlsadmin can establish a listener process for a given network, configure the specific attributes of that listener, and start and kill the listener process for that network. nlsadmin can also report on the listener processes on a machine, either individually (per network) or collectively.</p> <p><i>net_spec</i> represents a particular listener process. Specifically, <i>net_spec</i> is the relative path name of the entry under <i>/dev</i> for a given network (that is, a transport provider). <i>address</i> is a transport address on which to listen and is interpreted using a syntax that allows for a variety of address formats. By default, <i>address</i> is interpreted as the symbolic ASCII representation of the transport address. An <i>address</i> preceded by <i>\x</i> will let you enter an address in hexadecimal notation. Note that <i>address</i> must appear as a single word to the shell, thus it must be quoted if it contains any blanks.</p> <p>Changes to the list of services provided by the listener or the addresses of those services are put into effect immediately.</p>
OPTIONS	<p>nlsadmin may be used with the following combinations of options and arguments:</p> <ul style="list-style-type: none"> -x Report the status of all of the listener processes installed on this machine. net_spec Print the status of the listener process for <i>net_spec</i>. -q net_spec Query the status of the listener process for the specified network, and reflects the result of that query in its exit code. If a listener process is active, nlsadmin will exit with a status of 0; if no process is active, the exit code will be 1; the exit code will be greater than 1 in case of error.

- v** *net_spec* Print a verbose report on the servers associated with *net_spec*, giving the service code, status, command, and comment for each. It also specifies the **uid** the server will run as and the list of modules to be pushed, if any, before the server is started.
- z** *service_code net_spec* Print a report on the server associated with *net_spec* that has service code *service_code*, giving the same information as in the **-v** option.
- q -z** *service_code net_spec* Query the status of the service with service code *service_code* on network *net_spec*, and exits with a status of **0** if that service is enabled, **1** if that service is disabled, and greater than **1** in case of error.
- l** *address net_spec* Change or set the transport address on which the listener listens (the general listener service). This address can be used by remote processes to access the servers available through this listener (see the **-a** option, below).
If *address* is just a dash (-), **nlsadmin** reports the address currently configured, instead of changing it.
A change of address takes effect immediately.
- t** *address net_spec* Change or set the address on which the listener listens for requests for terminal service but is otherwise similar to the **-l** option above. A terminal service address should not be defined unless the appropriate remote login software is available; if such software is available, it must be configured as service code 1 (see the **-a** option, below).
- i** *net_spec* Initialize an instance of the listener for the network specified by *net_spec*; that is, create and initialize the files required by the listener as well as starting that instance of the listener. Note that a particular instance of the listener should be initialized only once. The listener must be initialized before assigning addresses or services.
- a** *service_code* [**-p** *modules*] [**-w** *name*] **-c** *cmd* **-y** *comment net_spec* Add a new service to the list of services available through the indicated listener. *service_code* is the code for the service, *cmd* is the command to be invoked in response to that service code, comprised of the full path name of the server and its arguments, and *comment* is a brief (free-form) description of the service for use in various reports. Note that *cmd* must appear as a single word to the shell; if arguments are required, the *cmd* and its arguments must be enclosed in quotation marks. The *comment* must also appear as a single word to the shell. When a service is added, it is initially enabled (see the **-e** and **-d** options, below).

Service codes are alphanumeric strings, and are administered by AT&T. The numeric service codes 0 through 100 are reserved for internal use by the listener. Service code 0 is assigned to the nlps server, which is the service invoked on the general listening address. In particular, code 1 is assigned to the remote login service, which is the service automatically invoked for connections to the terminal login address.

If the **-p** option is specified, then *modules* will be interpreted as a list of STREAMS modules for the listener to push before starting the service being added. The modules are pushed in the order they are specified. *modules* should be a comma-separated list of modules, with no white space included.

If the **-w** option is specified, then *name* is interpreted as the user name from `/etc/passwd` that the listener should look up. From the user name, the listener obtains the user ID, the group ID(s), and the home directory for use by the server. If **-w** is not specified, the default is to use the user name **listen**.

A service must explicitly be added to the listener for each network on which that service is to be available. This operation will normally be performed only when the service is installed on a machine, or when populating the list of services for a new network.

-r *service_code net_spec*

Remove the entry for the *service_code* from that listener's list of services. This is normally done only in conjunction with the de-installation of a service from a machine.

-e *service_code net_spec*

-d *service_code net_spec*

Enable or disable (respectively) the service indicated by *service_code* for the specified network. The service must previously have been added to the listener for that network (see the **-a** option, above). Disabling a service will cause subsequent service requests for that service to be denied, but the processes from any prior service requests that are still running will continue unaffected.

-s *net_spec*

-k *net_spec*

Start and kill (respectively) the listener process for the indicated network. These operations are normally performed as part of the system startup and shutdown procedures. Before a listener can be started for a particular network, it must first have been initialized (see the **-i** option, above). When a listener is killed, processes that are still running as a result of prior service requests will continue unaffected.

Under the Service Access Facility, it is possible to have multiple instances of the listener on a single *net_spec*. In any of the above commands, the option **-N** *port_monitor_tag* may be used in place of the *net_spec* argument. This argument specifies the tag by which an instance of the listener is identified by the Service Access Facility. If the **-N** option is not

specified (that is, the *net_spec* is specified in the invocation), then it will be assumed that the last component of the *net_spec* represents the tag of the listener for which the operation is destined. In other words, it is assumed that there is at least one listener on a designated *net_spec*, and that its tag is identical to the last component of the *net_spec*. This listener may be thought of as the primary, or default, listener for a particular *net_spec*.

nlsadmin is also used in conjunction with the Service Access Facility commands. In that capacity, the following combinations of options can be used:

-V Write the current version number of the listener's administrative file to the standard output. It is used as part of the **sacadm** command line when **sacadm** adds a port monitor to the system.

-c cmd | **-o streamname** [**-p modules**] [**-A address** | **-D**] [**-R prognum : versnum**]
Format the port monitor-specific information to be used as an argument to **pmadm**(1M).

The **-c** option specifies the full path name of the server and its arguments. *cmd* must appear as a single word to the shell, and its arguments must therefore be surrounded by quotes.

The **-o** option specifies the full path name of a FIFO or named STREAM through which a standing server is actually receiving the connection.

If the **-p** option is specified, then *modules* will be interpreted as a list of STREAMS modules for the listener to push before starting the service being added. The modules are pushed in the order in which they are specified. *modules* must be a comma-separated list, with no white space included.

If the **-A** option is specified, then *address* will be interpreted as the server's private address. The listener will monitor this address on behalf of the service and will dispatch all calls arriving on this address directly to the designated service. This option may not be used in conjunction with the **-D** option.

If the **-D** option is specified, then the service is assigned a private address dynamically, that is, the listener will have the transport provider select the address each time the listener begins listening on behalf of this service. For RPC services, this option will often be used in conjunction with the **-R** option to register the dynamically assigned address with the rpcbinder. This option may not be used in conjunction with the **-A** option.

When the **-R** option is specified, the service is an RPC service whose address, program number, and version number should be registered with the rpcbinder for this transport provider. This registration is performed each time the listener begins listening on behalf of the service. *prognum* and *versnum* are the program number and version number, respectively, of the RPC service.

nlsadmin may be invoked by any user to generate reports; all operations that affect a listener's status or configuration may only be run by a super-user.

The options specific to the Service Access Facility may not be used together with any other options.

- ERRORS** If successful, **nlsadmin** exits with a status of 0. If **nlsadmin** fails for any reason, it exits with a status greater than or equal to 2. See **-q** option for a return status of 1.
- SEE ALSO** **listen(1M)**, **pmadm(1M)**, **rpcbind(1M)**, **sacadm(1M)**
System Administration Guide, Volume II
- NOTES** Dynamically assigned addresses are not displayed in reports as statically assigned addresses are.

NAME	nscd – name service cache daemon
SYNOPSIS	<code>/usr/sbin/nscd [-f <i>configuration-file</i>] [-g] [-e <i>cachename, yes no</i>] [-i <i>cachename</i>]</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>nscd is a process that provides a cache for the most common name service requests. It is started up during multi-user boot. The default <i>configuration-file</i> <code>/etc/nscd.conf</code> determines the behavior of the cache daemon. See nscd.conf(4).</p> <p>nscd provides cacheing for the passwd(4), group(4) and hosts(4) databases through standard libc interfaces, such as gethostbyname(3N), gethostbyaddr(3N), and others. Each cache has a separate time-to-live for its data; modifying the local database (<code>/etc/hosts</code>, and so forth) causes that cache to become invalidated within ten seconds. Note that the shadow file is specifically not cached. getspnam(3C) calls remain uncached as a result.</p> <p>nscd also acts as its own administration tool. If an instance of nscd is already running, commands are passed to the running version transparently.</p> <p>In order to preserve NIS+ security, the startup script for nscd (<code>/etc/init.d/nscd</code>) checks the permissions on the passwd, group and host tables if NIS+ is being used. If those tables are not readable by unauthenticated users, then caching is disabled so that each process continues to authenticate itself as before.</p> <p>nscd does not rescan the <code>/etc/nsswitch.conf</code> file; if this file is changed, the machine should be rebooted or nscd stopped and restarted, as shown in the EXAMPLES below.</p>
OPTIONS	<p>Several of the options described below require a <i>cachename</i> specification. Supported values are passwd, group and hosts.</p> <p>-f <i>configuration-file</i> Causes nscd to read its configuration data from the specified file.</p> <p>-g Prints current configuration and statistics to standard output. This is the only option executable by non-root users.</p> <p>-e <i>cachename, yes no</i> Enables or disables the specified cache.</p> <p>-i <i>cachename</i> Invalidate the specified cache.</p>
EXAMPLES	<p>Stopping and restarting the nscd daemon.</p> <pre>example# /etc/init.d/nscd stop example# /etc/init.d/nscd start</pre>
FILES	<code>/etc/nscd.conf</code> determines behavior of cache daemon
SEE ALSO	gethostbyname (3N), group (4), hosts (4), nscd.conf (4), nsswitch.conf (4), passwd (4)

WARNINGS

The **nscd** interface is included in this release on an uncommitted basis only, and is subject to change or removal in a future minor release.

NAME	nslookup – query name servers interactively
SYNOPSIS	nslookup [-opt] [<i>host</i> / -] [<i>server</i>]
AVAILABILITY	SUNWcsu
DESCRIPTION	nslookup is an interactive program to query ARPA Internet domain name servers. The user can contact servers to request information about a specific host, or print a list of hosts in the domain.
OPTIONS	<p>-opt Allows you to set options as in the interactive set option shown below. For example -querytype=HINFO to ask for host information instead of the default query type of A for address information.</p> <p><i>host</i> Look up the host <i>host</i> directly and do not go into interactive mode.</p> <p>- Use the name server specified next on the command line instead of the servers in /etc/resolv.conf. Note that if both host and server are specified on the command line this - argument becomes optional.</p> <p><i>server</i> Use the name server specified. This can be either a name or an Internet address. If this fails, it will default to using the entries in the /etc/resolv.conf file.</p>
USAGE Overview	<p>The Internet domain name-space is tree-structured, with five top-level domains at present:</p> <p style="margin-left: 40px;">COM commercial establishments</p> <p style="margin-left: 40px;">EDU educational institutions</p> <p style="margin-left: 40px;">ORG not-for-profit organizations</p> <p style="margin-left: 40px;">GOV government agencies</p> <p style="margin-left: 40px;">MIL MILNET hosts</p> <p>If you are looking for a specific host, you need to know something about the host's organization in order to determine the top-level domain it belongs to. For instance, if you want to find the Internet address of a machine at UCLA , do the following:</p> <ul style="list-style-type: none"> • Connect with the root server using the root command. The root server of the name space has knowledge of the top-level domains. • Since UCLA is a university, its domain name is ucla.edu. Connect with a server for the ucla.edu domain with the command server ucla.edu. The response will print the names of hosts that act as servers for that domain. Note: the root server does not have information about ucla.edu, but knows the names and addresses of hosts that do. Once located by the root server, all future queries will be sent to the UCLA name server.

- To request information about a particular host in the domain (for instance, **locus**), just type the host name. To request a listing of hosts in the UCLA domain, use the **ls** command. The **ls** command requires a domain name (in this case, **ucla.edu**) as an argument.

If you are connected with a name server that handles more than one domain, all lookups for host names must be fully specified with its domain. For instance, the domain **harvard.edu** is served by **seismo.css.gov**, which also services the **css.gov** and **cornell.edu** domains. A lookup request for the host **aiken** in the **harvard.edu** domain must be specified as **aiken.harvard.edu**. However, the

set domain = name

and

set defname

commands can be used to automatically append a domain name to each request.

After a successful lookup of a host, use the **finger** command to see who is on the system, or to finger a specific person. To get other information about the host, use the

set querytype = value

command to change the type of information desired and request another lookup. (**finger** requires the type to be **A**.)

Commands

To exit, type CTRL-D (EOF).

The command line length must be less than 80 characters.

An unrecognized command will be interpreted as a host name.

host [*server*]

Look up information for *host* using the current default server, or using *server* if it is specified.

server *domain*

lserver *domain*

Change the default server to *domain*. **lserver** uses the initial server to look up information about *domain* while **server** uses the current default server. If an authoritative answer can not be found, the names of servers that might have the answer are returned.

root Change the default server to the server for the root of the domain name space.

Currently, the host **ns.nic.ddn.mil** is used; this command is a synonym for **lserver ns.nic.ddn.mil**. The name of the root server can be changed with the **set root** command.

finger [*name*]

Connect with the finger server on the current host, which is defined by a previous successful lookup for a host's address information (see the **set querytype = A** command). As with the shell, output can be redirected to a named file using **>** and **>>**.

ls [-ah]

List the information available for *domain*. The default output contains host names and their Internet addresses. The **-a** option lists aliases of hosts in the domain. The **-h** option lists CPU and operating system information for the domain. As with the shell, output can be redirected to a named file using **>** and **>>**. When output is directed to a file, hash marks are printed for every 50 records received from the server.

view filename

Sort and list the output of the **ls** command with **more(1)**.

help

? Print a brief summary of commands.

set keyword [= value] This command is used to change state information that affects the lookups. Valid keywords are:

all Print the current values of the various options to **set**. Information about the current default server and host is also printed.

[no]deb[ug]

Turn debugging mode on. A lot more information is printed about the packet sent to the server and the resulting answer. The default is **nodebug**.

[no]def[name]

Append the default domain name to every lookup. The default is **nodefname**.

do[main]= filename

Change the default domain name to *filename*. The default domain name is appended to all lookup requests if **defname** option has been set. The default is the value in **/etc/resolv.conf**.

q[querytype] = value

Change the type of information returned from a query to one of:

A	The host's Internet address (the default).
CNAME	The canonical name for an alias.
HINFO	The host CPU and operating system type.
MD	The mail destination.
MX	The mail exchanger.
MB	The mailbox domain name.
MG	The mail group member.
MINFO	The mailbox or mail list information.

(Other types specified in the RFC 883 document are valid, but are not very useful.)

[no]recurse

Tell the name server to query other servers if it does not have the information. The default is **recurse**.

ret[ry] = *count*

Set the number of times to retry a request before giving up to *count*.

When a reply to a request is not received within a certain amount of time (changed with **set timeout**), the request is resent. The default is *count* is 2.

ro[ot] = *host*

Change the name of the root server to *host*. This affects the **root** command. The default root server is **ns.nic.ddn.mil**.

t[timeout] = *interval*

Change the time-out for a reply to *interval* seconds. The default *interval* is 10 seconds.

[no]v[c]

Always use a virtual circuit when sending requests to the server. The default is **novc**.

FILES /etc/resolv.conf initial domain name and name server addresses

SEE ALSO nstest(1M), resolver(3N), resolv.conf(4)
RFC 882, RFC 883

DIAGNOSTICS If the lookup request was not successful, an error message is printed. Possible errors are:

Time-out The server did not respond to a request after a certain amount of time (changed with **set timeout = value**) and a certain number of retries (changed with **set retry = value**).

No information Depending on the query type set with the **set querytype** command, no information about the host was available, though the host name is valid.

Non-existent domain

The host or domain name does not exist.

Connection refused

Network is unreachable

The connection to the name or finger server could not be made at the current time. This error commonly occurs with **finger** requests.

Server failure The name server found an internal inconsistency in its database and could not return a valid answer.

Refused The name server refused to service the request.

The following error should not occur and it indicates a bug in the program.

Format error

The name server found that the request packet was not in the proper format.

NAME nstest – DNS test shell

SYNOPSIS nstest [**-d**] [**-i**] [**-r**] [**-v**] [**-p port**] [*inet_addr* [*logfile*]]

DESCRIPTION **nstest** is an interactive DNS test program. Queries are formed and sent by user command; any reply received is printed on the standard output. *inet_addr* is the Internet address of the DNS resolver to which **nstest** should send its queries. If *inet_addr* is not included, **nstest** first tries to contact a DNS server on the local host; if that fails, it tries the servers listed in the */etc/resolv.conf* file. If a *logfile* is supplied, **nstest** uses it to log the queries sent and replies received.

OPTIONS

- d** Causes **nstest** to create a file named **ns_packet.dump** (if it does not exist) and write into it a raw (binary) copy of each packet sent. If **ns_packet.dump** does exist, **nstest** will truncate it.
- i** Sets the **RES_IGNTC** flag on the queries it makes. See **resolver(3N)** for a description of the **RES_IGNTC** flag.
- r** Turns off the **RES_RECURSE** flag on the queries it makes. See **resolver(3N)** for a description of the **RES_RECURSE** flag.
- v** Turns on the **RES_USEVC** and **RES_STAYOPEN** flags on the **res_send ()** calls made. See **resolver(3N)** for a description of the **RES_USEVC** and **RES_STAYOPEN** flags.
- p** Causes **nstest** to use the supplied *port* instead of the default name server port.

USAGE When **nstest** starts, it prints a prompt ("**>**") and waits for user input. DNS queries are formed by typing a *key letter* followed by the appropriate *argument*. Each *key letter* results in a call to **res_mkquery ()** with *op* set to either **IQUERY** or **QUERY** and *type* set to one of the type values (defined in *<arpa/nameser.h>*). (Any other *key letter* than those listed below causes **nstest** to print a summary of the following table.)

Key Letter & Argument	Op	Type
<i>ahost</i>	QUERY	T_A
<i>Aaddr</i>	IQUERY	T_A
<i>Buser</i>	QUERY	T_MG
<i>buser</i>	QUERY	T_MB
<i>chost</i>	QUERY	T_CNAME
<i>fhost</i>	QUERY	T_UINFO
<i>Ggid</i>	IQUERY	T_GID
<i>ghost</i>	QUERY	T_GID
<i>hhost</i>	QUERY	T_HINFO
<i>ihost</i>	QUERY	T_MINFO
<i>Mhost</i>	QUERY	T_MAILB
<i>mhost</i>	QUERY	T_MX

<i>nhost</i>	QUERY	T_NS
<i>phost</i>	QUERY	T_PTR
<i>rhost</i>	QUERY	T_MR
<i>shost</i>	QUERY	T_SOA
<i>Thost</i>	QUERY	T_TXT
<i>Uuid</i>	IQUERY	T_UID
<i>uhost</i>	QUERY	T_UID
<i>whost</i>	QUERY	T_WKS
<i>xhost</i>	QUERY	T_AXFR

After the query is successfully formed, **res_send ()** is called to send it and wait for a reply. **nstest** then prints the following on the standard output:

- a summary of the request and reply packets, including the **HEADER** structure (defined in `<arpa/nameser.h>`) used in the request
- the question being asked of the name server
- an enumeration of the name server(s) being polled
- a summary of the **HEADER** structure received in the reply
- the question the name server answered
- the answer itself

EXAMPLES

To fetch the address of host **playground.sun.com** from the Sun name server, the user would enter:

```
$ nstest 192.9.5.1
> aplayground.sun.com
```

nstest would return the following:

```
res_mkquery(0, playground.sun.com, 1, 1)
res_send()
HEADER:
  opcode = QUERY, id = 1, rcode = NOERROR
  header flags: rd
  qdcount = 1, anccount = 0, nscount = 0, arcount = 0
```

```
QUESTIONS:
  playground.sun.com, type = A, class = IN
```

```
Querying server (# 1) address = 192.9.5.1
got answer:
HEADER:
  opcode = QUERY, id = 1, rcode = NOERROR
  header flags: qr aa rd ra
  qdcount = 1, anccount = 1, nscount = 0, arcount = 0
```

QUESTIONS:

playground.sun.com, type = A, class = IN

ANSWERS:

**playground.sun.com
type = A, class = IN, ttl = 1 day, dlen = 4
internet address = 192.9.5.5**

To look up a PTR record, enter:

```
$ nstest 192.9.5.1
> p5.5.9.192.in-addr.arpa
```

nstest would return the following:

```
res_mkquery(0, 5.5.9.192.in-addr.arpa, 1, 12)
res_send()
HEADER:
opcode = QUERY, id = 2, rcode = NOERROR
header flags: rd
qdcount = 1, ancound = 0, nscount = 0, arcount = 0
```

QUESTIONS:

5.5.9.192.in-addr.arpa, type = PTR, class = IN

Querying server (# 1) address = 192.9.5.1

got answer:

```
HEADER:
opcode = QUERY, id = 2, rcode = NOERROR
header flags: qr aa rd ra
qdcount = 1, ancound = 1, nscount = 0, arcount = 0
```

QUESTIONS:

5.5.9.192.in-addr.arpa, type = PTR, class = IN

ANSWERS:

**5.5.9.192.in-addr.arpa
type = PTR, class = IN, ttl = 7 hours 47 mins 2 secs, dlen = 23
domain name = playground.sun.com**

FILES **/usr/include/arpa/nameser.h** include file for implementation of DNS protocol
 /usr/include/resolv.h include file for the resolver daemon (in.named)

SEE ALSO **nslookup(1M), resolver(3N)**

NAME	obpsym – Kernel Symbolic Debugging for OpenBoot Firmware
SYNOPSIS	modload -p misc/obpsym
AVAILABILITY	SUNWcsr
DESCRIPTION	<p>obpsym is a kernel module that installs OpenBoot callback handlers that provide kernel symbol information to OpenBoot. OpenBoot firmware user interface commands use the callbacks to convert numeric <i>addresses</i> to kernel symbol names for display purposes, and to convert kernel symbol names to numeric <i>literals</i> allowing symbolic names to be used as input arguments to user interface commands.</p> <p>Once obpsym is installed, kernel symbolic names may be used anywhere at the OpenBoot firmware's user interface command prompt in place of a literal (numeric) string. For example, if obpsym is installed, the OpenBoot firmware commands ctrace and dis typically display symbolic names and offsets in the form <i>modname:symbolname + offset</i>. User interface Commands such as dis can be given a kernel symbolic name such as ufs:ufs_mount instead of a numeric address.</p> <p>Placing the command</p> <p style="padding-left: 40px;">forceload: misc/obpsym</p> <p>into the system(4) file forces the kernel module misc/obpsym to be loaded and activates the kernel callbacks during the kernel startup sequence.</p> <p>obpsym may be useful as a kernel debugger in situations where other kernel debuggers are not useful. For example, on SPARC machines, if obpsym is loaded, you may be able to use the OpenBoot firmware's ctrace command to display symbolic names in the stack backtrace after a watchdog reset.</p>
Kernel Symbolic Name Syntax	<p>The syntax for a kernel symbolic name is:</p> <p style="padding-left: 40px;">[<i>module-name</i> :] <i>symbol-name</i></p> <p>Where <i>module-name</i> is the name of the kernel module that the symbol <i>symbol-name</i> appears in. A NULL module name is taken as "all modules, in no particular order" by obpsym. The module name unix is equivalent to a NULL module name, so that conflicts with words defined in the firmware's vocabulary can be avoided.</p> <p>Typically, OpenBoot firmware reads a word from the input stream and looks the word up in its internal <i>vocabulary</i> before checking if the word is a <i>literal</i>. Thus, kernel symbols, such as reset may be given as unix:reset to avoid the unexpected side effect of the firmware finding and executing a matching word in its vocabulary.</p>
FILES	<p>/etc/system system configuration information file</p> <p>/platform/platform-name/kernel/misc/obpsym</p>

SEE ALSO **kadb(1M)**, **kernel(1M)**, **modload(1M)**, **modunload(1M)**, **system(4)**, **uname(1)**
OpenBoot Command Reference, Version 2.x, 1994, SunSoft Press, ISBN Number 0-13-107194-7

WARNINGS Some OpenBoot firmware user interface commands may use system resources incompatibly with the way they are used by the Unix kernel. These commands and the use of this feature as a kernel debugger may cause interactions that the Unix kernel is not prepared to deal with. If this occurs, the Unix kernel and/or the OpenBoot firmware user interface commands may react unpredictably and may panic the system, or may hang or may cause other unpredictable results. For these reasons, the use of this feature is only minimally supported and recommended to be used only as a kernel debugger of "last resort".

NOTES *platform-name* can be found using the **-i** option of **uname(1)**
obpsym is supported only on architectures that support OpenBoot firmware.
On some systems, OpenBoot must be completely RAM resident so the **obpsym** symbol callback support can be added to the firmware, if the firmware doesn't include support for the symbol callbacks. On these systems, **obpsym** may complain that it requires that "you must use ramforth to use this module".
See the **OpenBoot Command Reference** manual for details on how to use the *ramforth* command, how to place the command into *nvrarc*, and how to set *use-nvrarc?* to *true*. On systems with version 1.x OpenBoot firmware, *nvrarc* doesn't exist, and the *ramforth* command must be typed manually after each reset, in order to use this module.
Once installed, the symbol table callbacks can be disabled by using the following OpenBoot firmware command:

0 0 set-symbol-lookup

NAME	parse_dynamic_clustertoc – parse clustertoc file based on dynamic entries
SYNOPSIS	parse_dynamic_clustertoc
AVAILABILITY	SHWPCdrom (Solaris CD)
DESCRIPTION	<p>This script parses the clustertoc file before the suninstall(1M) process is run. parse_dynamic_clustertoc is called by a modified sysconfig script on the install CD. When parse_dynamic_clustertoc runs, it reads the clustertoc and when it encounters SUNW_CSRMBRIFF lines, it either checks the platform using the script's builtin function, or calls an external script. The script exits with a 0 if the cluster entry is included, otherwise it will be ignored. If the cluster entry is to be included, the SUNW_CSRMBRIFF =(<i><test></i> <i><test_arg></i>)<i><cluster></i> line is converted to SUNW_CSRMEMBER =<i><cluster></i>.</p>
EXAMPLES	<p>The following is an example of a simple external test to check for a SX Framebuffer. The entry in the clustertoc file is shown and following that is the script that would be placed in the install.d/dynamic_test directory.</p> <pre> SUNW_CSRMBRIFF=(smcc.dctoc sx)SUNWCsx #!/bin/sh # # Likewise, this file is expected to live under \$(TESTDIR). # case "\$1" in sx) prtconf -p grep `SUNW,sx` 1> /dev/null;; esac </pre>
FILES	<p><cdrom>/Solaris_2.x/locale/C/.clustertoc.dynamic dynamic version of the clustertoc file</p> <p><cdrom>/export/exec/sparc.Solaris_2.x/sbin/install.d/dynamic_test directory that contains any additional tests</p>
SEE ALSO	suninstall(1M) , clustertoc(4)

NAME	passmgmt – password files management
SYNOPSIS	<p>passmgmt –<i>a options name</i> passmgmt –<i>m options name</i> passmgmt –<i>d name</i></p>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>The passmgmt command updates information in the password files. This command works with both /etc/passwd and /etc/shadow.</p> <p>passmgmt –<i>a</i> adds an entry for user <i>name</i> to the password files. This command does not create any directory for the new user and the new login remains locked (with the string *LK* in the password field) until the passwd(1) command is executed to set the password.</p> <p>passmgmt –<i>m</i> modifies the entry for user <i>name</i> in the password files. The name field in the /etc/shadow entry and all the fields (except the password field) in the /etc/passwd entry can be modified by this command. Only fields entered on the command line will be modified.</p> <p>passmgmt –<i>d</i> deletes the entry for user <i>name</i> from the password files. It will not remove any files that the user owns on the system; they must be removed manually.</p> <p>passmgmt can be used only by the super-user.</p>
OPTIONS	<p>–<i>c comment</i> A short description of the login, enclosed in quotes. It is limited to a maximum of 128 characters and defaults to an empty field.</p> <p>–<i>h homedir</i> Home directory of <i>name</i>. It is limited to a maximum of 256 characters and defaults to /usr/home.</p> <p>–<i>u uid</i> UID of the <i>name</i>. This number must range from 0 to the maximum non-negative value for the system. It defaults to the next available UID greater than 99. Without the –<i>o</i> option, it enforces the uniqueness of a UID.</p> <p>–<i>o</i> This option allows a UID to be non-unique. It is used only with the –<i>u</i> option.</p> <p>–<i>g gid</i> GID of the <i>name</i>. This number must range from 0 to the maximum non-negative value for the system. The default is 1.</p> <p>–<i>s shell</i> Login shell for <i>name</i>. It should be the full pathname of the program that will be executed when the user logs in. The maximum size of <i>shell</i> is 256 characters. The default is for this field to be empty and to be interpreted as /usr/bin/sh.</p> <p>–<i>l logname</i> This option changes the <i>name</i> to <i>logname</i>. It is used only with the –<i>m</i> option. The total size of each login entry is limited to a maximum of 511 bytes in each of the password files.</p>

FILES /etc/passwd
/etc/shadow
/etc/passwd
/etc/shadow

SEE ALSO passwd(1), useradd(1M), userdel(1M), usermod(1M), passwd(4), shadow(4)

DIAGNOSTICS The **passmgmt** command exits with one of the following values:

- 0 Success.
- 1 Permission denied.
- 2 Invalid command syntax. Usage message of the **passmgmt** command is displayed.
- 3 Invalid argument provided to option.
- 4 UID in use.
- 5 Inconsistent password files (for example, *name* is in the **/etc/passwd** file and not in the **/etc/shadow** file, or vice versa).
- 6 Unexpected failure. Password files unchanged.
- 7 Unexpected failure. Password file(s) missing.
- 8 Password file(s) busy. Try again later.
- 9 *name* does not exist (if **-m** or **-d** is specified), already exists (if **-a** is specified), or *logname* already exists (if **-m -l** is specified).

NOTES You cannot use a colon or carriage return as part of an argument because it is interpreted as a field separator in the password file.

This command will be removed in a future release. Its functionality has been replaced and enhanced by **useradd**, **userdel**, and **usermod**. These commands are currently available.

This command only modifies password definitions in the local **/etc/passwd** and **/etc/shadow** files. If a network nameservice such as **NIS** or **NIS+** is being used to supplement the local files with additional entries, **passmgmt** cannot change information supplied by the network nameservice.

NAME	pbind – control and query bindings of processes to processors
SYNOPSIS	pbind -b <i>processor_id</i> <i>pid</i> ... pbind -u <i>pid</i> ... pbind [-q] [<i>pid</i> ...]
AVAILABILITY	SUNWcsu
DESCRIPTION	pbind binds all the LWPs (light-weight processes) of a process to a processor, or removes or displays the bindings.
OPTIONS	-b <i>processor_id</i> binds all the LWPs of the specified processes to the processor <i>processor_id</i> . -u removes the bindings of all LWPs of the specified processes. -q Displays the bindings of the specified processes, or of all processes.
USAGE	The -b option binds all of the LWPs of the specified processes to the specified processor. The processor must be present and on-line, which can be determined by the psrinfo(1M) command. When an LWP is bound to a processor, it will be executed only by that processor except when the LWP requires a resource that is provided only by another processor. The binding is not exclusive, that is, the processor is free execute other LWPs as well. Bindings are inherited, so new LWPs and processes created by a bound LWP will have the same binding. Binding an interactive shell to a processor, for example, binds all commands executed by the shell. The superuser may bind or unbind any process, and other users can use pbind to bind or unbind any process for which the user has permission to signal. That is, any process that has the same effective user ID as the user. The -u option removes the bindings from all the LWPs of the specified processes, allowing them to be executed on any on-line processor. The -q option displays the bindings of the specified processes. If a process is composed of multiple LWPs, which have different bindings, the bindings of only one of the bound LWPs will be shown.
EXAMPLES Binding processes	The following command binds processes 204 and 223 to processor 2: <pre>example% pbind -b 2 204 223</pre> This generates the messages: <pre>process id 204: was 2, now 2 process id 223: was 3, now 2</pre>

Unbinding a process

To unbind process 204:

```
example% pbind -u 204
```

Querying Bindings

The command

```
example% pbind -q 1 149 101
```

generates the following output:

```
process id 1: 0
```

```
process id 149: 3
```

```
process id 101: not bound
```

This example demonstrates that process 1 is bound to processor 0, process 149 has at least one LWP bound to CPU3, and process 101 has no bound LWPs.

SEE ALSO

psrinfo(1M), **psradm(1M)**, **processor_bind(2)**, **processor_info(2)**, **sysconf(3C)**

DIAGNOSTICS

pbind: cannot query pid 31: No such process

The process specified did not exist or has exited.

pbind: cannot bind pid 31: Not owner

The user does not have permission to bind the process.

pbind: cannot bind pid 31: Invalid argument

The specified processor is offline.

NAME	pcmcia – PCMCIA user daemon
SYNOPSIS	/usr/lib/pcmcia
AVAILABILITY	SUNWpcmcu
DESCRIPTION	The PCMCIA user daemon provides user-level services for the PCMCIA nexus driver and PCMCIA card client drivers. There are no user-configurable options for this daemon.
SEE ALSO	pcmcia(4)
DIAGNOSTICS	pcmcia: can't open /dev/pem: No such file or directory The user daemon could not communicate with the PCMCIA event management driver.

NAME	pfinstall – tests installation profiles
SYNOPSIS	<code>/usr/sbin/install.d/pfinstall -D -d disk_config [-c path] profile [rootdisk]</code>
DESCRIPTION	<p>When the initial_install argument is defined in a profile, you can use the pfinstall command to test the profile without performing an actual installation. pfinstall enables you to test a profile by using either:</p> <ul style="list-style-type: none"> • The disk configuration on the system where pfinstall is being run. • A disk configuration file created by the prvtoc(1M) command. For example, disk configuration files enable you to test profiles for all the different systems on your network from one system. <p>pfinstall is available only on a system running the Solaris 2.x environment.</p>
OPTIONS	<p>-D pfinstall uses the system's disk configuration to test the profile. You must be root to execute the pfinstall command with the -D option.</p> <p>-d disk_config pfinstall uses a disk configuration file, <i>disk_config</i>, to test the profile. A disk configuration file can be created by using the prvtoc(1M) command.</p> <p>-c path The path to the Solaris 2.x installation image. This is required if the image is not mounted on /cdrom. (For example, use this option if you copied the installation image to disk or mounted the CD-ROM on a directory other than /cdrom.)</p> <p><i>profile</i> The file name of the profile to test.</p> <p><i>rootdisk</i> A disk's device name that specifies where to install the root (/) file system. Must be in the form cXtXdX.</p> <p>You should run pfinstall from the directory where the <i>profile</i> and <i>disk_config</i> files reside (such as the profile directory). If the <i>profile</i> or <i>disk_config</i> file is not in the directory where pfinstall is run, you must specify the path to the particular file.</p>
EXAMPLES	<p>The following examples use pfinstall to test the marketing profile:</p> <pre>example% /usr/sbin/install.d/pfinstall -D marketing example% /usr/sbin/install.d/pfinstall -d 104_test marketing example% /usr/sbin/install.d/pfinstall -D -c /export/install marketing</pre>
SEE ALSO	prvtoc(1M)
WARNINGS	If the -d or -D option is not specified, pfinstall will perform an actual installation on the system by using the specified profile, and the data on the system will be overwritten.

NOTES

To create a disk configuration file:

1. Locate a system with the disk size needed for testing a particular profile.
2. Determine the device name for the system's disk.
3. Redirect the output of **prtvtoc(1M)** to create the disk configuration file:

```
example% prtvtoc /dev/rdisk/device_name > disk_config
```

Where

/dev/rdisk/device_name is the device name of the system's disk (in the form **cXtXdXs2**)

disk_config is the disk configuration file name.

The following example creates a disk configuration file, **104_test**, on a system with a 104-Mbyte disk, whose device name is **c0t3d0**:

```
example% prtvtoc /dev/rdisk/c0t3d0s2 > 104_test
```

You can also create a disk configuration file that includes multiple disks. In the following example, a disk configuration file is created that includes two, 104-Mbyte disks. When creating disk configuration files with multiple disks, it is important to make sure that each disk in the file is specified with a different target.

```
example% cat 104_test 104_test > dual104_test
```

To test a profile with a specific system memory size, set **SYS_MEMSIZE** to the specific memory size (in Mbytes) before running **pfinstall**:

```
# SYS_MEMSIZE=memory_size
```

```
# export SYS_MEMSIZE
```

NAME	ping – send ICMP ECHO_REQUEST packets to network hosts
SYNOPSIS	<pre>/usr/sbin/ping host [timeout] /usr/sbin/ping [-s] [-dLLnrRv] [-i interface] [-I interval] [-t ttl] host [packetsize] [count]</pre>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>ping utilizes the ICMP protocol's ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from the specified <i>host</i> or network gateway. If <i>host</i> responds, ping will print host is alive on the standard output and exit. Otherwise after <i>timeout</i> seconds, it will write no answer from host. The default value of <i>timeout</i> is 20 seconds.</p> <p>When the -s flag is specified, ping sends one datagram per second (adjustable with -I), and prints one line of output for every ECHO_RESPONSE that it receives. No output is produced if there is no response. In this second form, ping computes round trip times and packet loss statistics; it displays a summary of this information upon termination or timeout. The default datagram packet size is 64 bytes, or you can specify a size with the <i>packetsize</i> command-line argument. If an optional <i>count</i> is given, ping sends only that number of requests.</p> <p>When using ping for fault isolation, first ping the local host to verify that the local network interface is running.</p> <p>If ping is successful and the host responds, the exist status is 0. If a host does not respond, or an error was returned, the exit status is 1.</p>
OPTIONS	<ul style="list-style-type: none"> -d Set the SO_DEBUG socket option. -I Loose source route. Use this option in the IP header to send the packet to the given host and back again. Usually specified with the -R option. -L Turn off loopback of multicast packets. Normally, if there are members in the host group on the outgoing interface, a copy of the multicast packets will be delivered to the local machine. -n Show network addresses as numbers. ping normally displays addresses as host names. -r Bypass the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has been dropped by the router daemon (see in.routed(1M)). -R Record route. Sets the IP record route option, which will store the route of the packet inside the IP header. The contents of the record route will only be printed if the -v option is given, and only be set on return packets if the target host preserves the record route option across echos, or the -I option is given. -v Verbose output. List any ICMP

packets, other than **ECHO_RESPONSE**, that are received.

-i interface

Specify the outgoing interface to use for multicast packets. The default interface for multicast packets is determined from the (unicast) routing tables.

-I interval

Specify the interval between successive transmissions. The default is one second.

-t ttl

Specify the IP time to live for unicast and multicast packets. The default time to live for unicast packets is set with `ndd` (using the **icmp_def_ttl** variable). The default time to live for multicast is one hop.

EXIT STATUS

The following exit values are returned:

0 Success — the machine is alive.

non-zero An error has occurred — either a malformed argument has been specified, or the machine was not alive.

SEE ALSO

ifconfig(1M), **in.routed(1M)**, **netstat(1M)**, **ndd(1M)**, **rpcinfo(1M)**, **icmp(7P)**

NAME	pkgadd – transfer software package to the system
SYNOPSIS	<pre>pkgadd [-n] [-a <i>admin</i>] [-d <i>device</i>] [-R <i>root_path</i>] [-r <i>response</i>] [<i>pkginst1</i> [<i>pkginst2</i>] ...] pkgadd -s <i>spool</i> [-d <i>device</i>] [<i>pkginst1</i> [<i>pkginst2</i>] ...]</pre>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>pkgadd transfers the contents of a software package from the distribution medium or directory to install it onto the system. Used without the -d option, pkgadd looks in the default spool directory for the package (/var/spool/pkg). Used with the -s option, it reads the package to a spool directory instead of installing it.</p> <p>Certain unbundled and third-party packages are no longer entirely compatible with the latest version of pkgadd. These packages require user interaction throughout the installation and not just at the very beginning.</p> <p>To install these older packages (released prior to Solaris 2.4), set the following environment variable:</p> <pre style="text-align: center;">NONABI_SCRIPTS=TRUE</pre> <p>pkgadd will permit keyboard interaction throughout the installation as long as this environment variable is set.</p>
OPTIONS	<p>-n Installation occurs in non-interactive mode. The default mode is interactive.</p> <p>-a <i>admin</i> Define an installation administration file, <i>admin</i>, to be used in place of the default administration file. The token none overrides the use of any <i>admin</i> file, and thus forces interaction with the user. Unless a full path name is given, pkgadd first looks in the current working directory for the administration file. If the specified administration file is not in the current working directory, pkgadd looks in the /var/sadm/install/admin directory for the administration file.</p> <p>-d <i>device</i> Install or copy a package from <i>device</i>. <i>device</i> can be a full path name to a directory or the identifiers for tape, floppy disk or removable disk (for example, /var/tmp or /floppy/floppy_name). It can also be a device alias (for example, /floppy/floppy0).</p> <p>-R <i>root_path</i> Define the full path name of a subdirectory to use as the <i>root_path</i>. All files, including package system information files, are relocated to a directory tree starting in the specified <i>root_path</i>.</p> <p>-r <i>response</i> Identify a file or directory which contains output from a previous pkgask session. This file supplies the interaction responses that would be requested by the package in interactive mode. <i>response</i> must be a full pathname.</p> <p>-s <i>spool</i> Reads the package into the directory <i>spool</i> instead of installing it.</p>

pkginst Specify the package instance or list of instances to be installed. The token **all** may be used to refer to all packages available on the source medium. The format *pkginst.** can be used to indicate all instances of a package.

When executed without options, **pkgadd** uses **/var/spool/pkg** (the default spool directory).

EXAMPLES

The following example installs a package from a Solaris CD-ROM. You are prompted for the name of the package you want to install.

```
example% pkgadd -d /cdrom/solaris_2_3/s0/Solaris_2.3
```

EXIT STATUS

The following error values are returned:

0 Successful completion.

>0 An error occurred.

SEE ALSO

pkginfo(1), **pkgmk(1)**, **pkgparam(1)**, **pkgproto(1)**, **pkgtrans(1)**, **installf(1M)**, **pkgask(1M)**, **pkgrm(1M)**, **removef(1M)**, **admin(4)**

NOTES

When transferring a package to a spool directory, the **-r**, **-n**, and **-a** options cannot be used.

The **-r** option can be used to indicate a directory name as well as a filename. The directory can contain numerous *response* files, each sharing the name of the package with which it should be associated. This would be used, for example, when adding multiple interactive packages with one invocation of **pkgadd**.

Each package would need a *response* file. If you create *response* files with the same name as the package (that is, *pkinst1* and *pkinst2*), then name the directory in which these files reside after the **-r**.

The **-n** option causes the installation to halt if any interaction is needed to complete it.

NAME	pkgask – stores answers to a request script
SYNOPSIS	pkgask [-d <i>device</i>] [-R <i>root_path</i>] -r <i>response</i> <i>pkginst1</i> [<i>pkginst2</i>] ...]
AVAILABILITY	SUNWcsu
DESCRIPTION	pkgask allows the administrator to store answers to an interactive package (one with a request script). Invoking this command generates a <i>response</i> file that is then used as input at installation time. The use of this <i>response</i> file prevents any interaction from occurring during installation since the file already contains all of the information the package needs.
OPTIONS	<p>-d <i>device</i> Run the request script for a package on <i>device</i>. <i>device</i> can be a directory path-name or the identifiers for tape, floppy disk or removable disk (for example, <i>/var/tmp</i>, <i>/dev/diskette</i>, and <i>/dev/dsk/c1d0s0</i>). The default device is the installation spool directory.</p> <p>-R <i>root_path</i> Define the full path name of a subdirectory to use as the <i>root_path</i>. All files, including package system information files, are relocated to a directory tree starting in the specified <i>root_path</i>.</p> <p>-r <i>response</i> Identify a file or directory which should be created to contain the responses to interaction with the package. The name must be a full pathname. The file, or directory of files, can later be used as input to the pkgadd command.</p> <p><i>pkginst</i> Specify the package instance, or list of instances for which request scripts will be created. The token all may be used to refer to all packages available on the source medium.</p>
SEE ALSO	pkginfo(1) , pkgmk(1) , pkgparam(1) , pkgproto(1) , pkgtrans(1) , installf(1M) , pkgadd(1M) , pkgchk(1M) , pkgrm(1M) , removef(1M)
NOTES	The -r option can be used to indicate a directory name as well as a filename. The directory name is used to create numerous <i>response</i> files, each sharing the name of the package with which it should be associated. This would be used, for example, when you will be adding multiple interactive packages with one invocation of pkgadd . Each package would need a <i>response</i> file. To create multiple response files with the same name as the package instance, name the directory in which the files should be created and supply multiple instance names with the pkgask command. When installing the packages, you will be able to identify this directory to the pkgadd command.

NAME	pkgchk – check accuracy of installation
SYNOPSIS	<pre>pkgchk [-l -acfnqv] [-p path [, ...]] [-i file] [[pkginst [...]] [-m pkgmap [-e envfile]]] pkgchk -d device [-f] [-l -v] [-p path [, ...]] [-i file] [pkginst[...]]</pre>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>pkgchk checks the accuracy of installed files or, by using the -l option, displays information about package files. pkgchk checks the integrity of directory structures and files. Discrepancies are written to standard error along with a detailed explanation of the problem.</p> <p>The first synopsis defined above is used to list or check the contents and/or attributes of objects that are currently installed on the system, or in the indicated <i>pkgmap</i>. Package names may be listed on the command line, or by default, the entire contents of a machine will be checked.</p> <p>The second synopsis is used to list or check the contents of a package which has been spooled on the specified device, but not installed. Note that attributes cannot be checked for spooled packages.</p>
OPTIONS	<p>The following options are supported:</p> <ul style="list-style-type: none"> -a Audit the file attributes only and do not check file contents. Default is to check both. -c Audit the file contents only and do not check file attributes. Default is to check both. -d device Specify the device on which a spooled package resides. <i>device</i> can be a directory path name or the identifiers for tape, floppy disk or removable disk (for example, /var/tmp or /dev/diskette). -e envfile Request that the pkginfo file named as <i>envfile</i> be used to resolve parameters noted in the specified <i>pkgmap</i> file. -f Correct file attributes if possible. If used with the -x option, this option removes hidden files. When pkgchk is invoked with this option it creates directories, named pipes, links and special devices if they do not already exist. If the -d option calls out an uninstalled package, the -f option will only take effect if the package is in directory (not stream) format. All file attributes will be set to agree with the entries in the <i>pkgmap</i> file except that setuid, setgid and sticky bits will not be set in the mode. -i file Read a list of path names from <i>file</i> and compares this list against the installation software database or the indicated <i>pkgmap</i> file. Path names which are not contained in <i>file</i> are not checked. -l List information on the selected files that make up a package. This option is not compatible with the -a, -c, -f, -g, and -v options.

- m** *pkgmap* Check the package against the package map file *pkgmap*.
- n** Do not check volatile or editable files. This should be used for most post-installation checking.
- p** *path* Only check the accuracy of the path name or path names listed. *path* can be one or more path names separated by commas (or by white space, if the list is quoted).
- q** Quiet mode. Do not give messages about missing files.
- v** Verbose mode. Files are listed as processed.
- x** Search exclusive directories, looking for files which exist that are not in the installation software database or the indicated *pkgmap* file.
- pkginst* Specify the package instance or instances to be checked. The format *pkginst.** can be used to check all instances of a package. The default is to display all information about all installed packages.

EXIT STATUS The following exit values are returned:

- 0** success.
- 1** error has occurred.

SEE ALSO **pkginfo(1), pkgtrans(1), pkgadd(1M), pkgask(1M), pkgrm(1M)**

NAME	pkgrm – removes a package from the system
SYNOPSIS	pkgrm [-n] [-R <i>root_dir</i>] [-a <i>admin</i>] [<i>pkginst1</i> [<i>pkginst2</i>] ...] pkgrm -s <i>spool</i> [<i>pkginst</i>]
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>pkgrm will remove a previously installed or partially installed package from the system. A check is made to determine if any other packages depend on the one being removed. If a dependency exists, the action taken is defined in the admin file.</p> <p>The default state for the command is in interactive mode, meaning that prompt messages are given during processing to allow the administrator to confirm the actions being taken. Non-interactive mode can be requested with the -n option.</p> <p>The -s option can be used to specify the directory from which spooled packages should be removed.</p> <p>Certain unbundled and third-party packages are no longer entirely compatible with the latest version of pkgrm. These packages require user interaction throughout the removal and not just at the very beginning.</p> <p>To remove these older packages (released prior to Solaris 2.4), set the following environment variable:</p> <p style="text-align: center;">NONABI_SCRIPTS=TRUE</p> <p>pkgrm will permit keyboard interaction throughout the removal as long as this environment variable is set.</p>
OPTIONS	<p>-n Non-interactive mode. If there is a need for interaction, the command will exit. Use of this option requires that at least one package instance be named upon invocation of the command.</p> <p>-R Defines the full path name of a subdirectory to use as the <i>root_path</i>. All files, including package system information files, are relocated to a directory tree starting in the specified <i>root_path</i>.</p> <p>-a <i>admin</i> Use the installation administration file, <i>admin</i>, in place of the default <i>admin</i> file. pkgrm first looks in the current working directory for the administration file. If the specified administration file is not in the current working directory, pkgrm looks in the /var/sadm/install/admin directory for the administration file.</p> <p>-s <i>spool</i> Remove the specified package(s) from the directory <i>spool</i>. The default directory for spooled packages is /var/sadm/spool/okg.</p> <p><i>pkginst</i> Specifies the package to be removed. The format <i>pkginst.*</i> can be used to remove all instances of a package.</p>

SEE ALSO

**pkginfo(1), pkgmk(1), pkgparam(1), pkgproto(1), pkgtrans(1), installf(1M),
pkgadd(1M), pkgask(1M), pkgchk(1M), removef(1M)**

NAME	pmadm – port monitor administration
SYNOPSIS	<pre>pmadm -a [-p pmtag -t type] -s svctag -i id -m pmspecific -v ver [-f xu] [-y comment] [-z script] pmadm -r -p pmtag -s svctag pmadm -e -p pmtag -s svctag pmadm -d -p pmtag -s svctag pmadm -l [-t type -p pmtag] [-s svctag] pmadm -L [-t type -p pmtag] [-s svctag] pmadm -g -p pmtag -s svctag [-z script] pmadm -g -s svctag -t type -z script</pre>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>pmadm is the administrative command for the lower level of the Service Access Facility hierarchy, that is, for service administration. A port may have only one service associated with it although the same service may be available through more than one port. In order to uniquely identify an instance of a service, the pmadm command must identify both the port monitor or port monitors through which the service is available (-p or -t) and the service (-s). See the option descriptions below.</p> <p>pmadm performs the following functions:</p> <ul style="list-style-type: none"> • add or remove a service • enable or disable a service • install or replace a per-service configuration script • print requested service information <p>Any user on the system may invoke pmadm to request service status (-l or -L) or to print per-service configuration scripts (-g without the -z option). pmadm with other options may be executed only by a privileged user.</p>
OPTIONS	<p>-a Add a service. pmadm adds an entry for the new service to the port monitor's administrative file. Because of the complexity of the options and arguments that follow the -a option, it may be convenient to use a command script or the menu system to add services.</p> <p>-d Disable a service. Add x to the flag field in the entry for the service <i>svctag</i> in the port monitor's administrative file. This is the entry used by port monitor <i>pmtag</i>. See the -f option, below, for a description of the flags available.</p> <p>-e Enable a service. Remove x from the flag field in the entry for the service <i>svctag</i> in the port monitor administrative file. This is the entry used by port monitor <i>pmtag</i>. See the -f option, below, for a description of the flags available.</p>

- f xu** The **-f** option specifies one or both of the following two flags which are then included in the flag field of the entry for the new service in the port monitor's administrative file. If the **-f** option is not included, no flags are set and the default conditions prevail. By default, a new service is enabled and no **utmp** entry is created for it. A **-f** option without a following argument is illegal.
- x** Do not enable the service *svctag* available through port monitor *pmtag*.
 - u** Create a **utmp** entry for service *svctag* available through port monitor *pmtag*.
- g** Print, install, or replace a per-service configuration script. The **-g** option with a **-p** option and a **-s** option prints the per-service configuration script for service *svctag* available through port monitor *pmtag*. The **-g** option with a **-p** option, a **-s** option, and a **-z** option installs the per-service configuration script contained in the file *script* as the per-service configuration script for service *svctag* available through port monitor *pmtag*. The **-g** option with a **-s** option, a **-t** option, and a **-z** option installs the file *script* as the per-service configuration script for service *svctag* available through any port monitor of type *type*. Other combinations of options with **-g** are invalid.
- i id** *id* is the identity that is to be assigned to service *svctag* when it is started. *id* must be an entry in */etc/passwd*.
- l** The **-l** option requests service information. Used by itself and with the options described below it provides a filter for extracting information in several different groupings.
- l** By itself, the **-l** option lists all services on the system.
 - l -p pmtag** Lists all services available through port monitor *pmtag*.
 - l -s svctag** Lists all services with tag *svctag*.
 - l -p pmtag-ssvctag**
Lists service *svctag*.
 - l -t type** Lists all services available through port monitors of type *type*.
 - l -t type-ssvctag**
Lists all services with tag *svctag* available through a port monitor of type *type*.
- Other combinations of options with **-l** are invalid.
- L** The **-L** option is identical to the **-l** option except that output is printed in a condensed format.
- m pmspecific**
pmspecific is the port monitor-specific portion of the port monitor administrative file entry for the service.

- p *pmtag*** Specifies the tag associated with the port monitor through which a service (specified as **-s *svctag***) is available.
- r** Remove a service. When **pmadm** removes a service, the entry for the service is removed from the port monitor's administrative file.
- s *svctag*** Specifies the service tag associated with a given service. The service tag is assigned by the system administrator and is part of the entry for the service in the port monitor's administrative file.
- t *type*** Specifies the the port monitor type.
- v *ver*** Specifies the version number of the port monitor administrative file. The version number may be given as
 -v '*pmspec* -V'
 where *pmspec* is the special administrative command for port monitor *pmtag*. This special command is **tyadm** for **tymon** and **nlsadmin** for **listen**. The version stamp of the port monitor is known by the command and is returned when *pmspec* is invoked with a **-V** option.
- y *comment***
 Associate *comment* with the service entry in the port monitor administrative file.
- z *script*** Used with the **-g** option to specify the name of the file that contains the per-service configuration script. Modifying a configuration script is a three-step procedure. First a copy of the existing script is made (**-g** alone). Then the copy is edited. Finally, the copy is put in place over the existing script (**-g** with **-z**).

Options that request information write the requested information to the standard output. A request for information using the **-I** option prints column headers and aligns the information under the appropriate headings. In this format, a missing field is indicated by a hyphen. A request for information in the condensed format using the **-L** option prints the information in colon-separated fields; missing fields are indicated by two successive colons. # is the comment character.

EXAMPLES

Add a service to a port monitor with tag **pmtag**. Give the service the tag **svctag**. Port monitor-specific information is generated by **specpm**. The service defined by **svctag** will be invoked with identity **root**.

```
pmadm -a -p pmtag -s svctag -i root -m `specpm -a arg1 -b arg2`
      -v `specpm -V`
```

Add a service with service tag **svctag**, identity **guest**, and port monitor-specific information generated by **specpm** to all port monitors of type **type**:

```
pmadm -a -s svctag -i guest -t type -m `specpm -a arg1 -b arg2`
      -v `specpm -V`
```

Remove the service **svctag** from port monitor **pmtag**:

```
pmadm -r -p pmtag -s svctag
```

Enable the service **svctag** available through port monitor **pmtag**:

```
pmadm -e -p pmtag -s svctag
```

Disable the service **svctag** available through port monitor **pmtag**:

```
pmadm -d -p pmtag -s svctag
```

List status information for all services:

```
pmadm -l
```

List status information for all services available through the port monitor with tag **ports**:

```
pmadm -l -p ports
```

List the same information in condensed format:

```
pmadm -L -p ports
```

List status information for all services available through port monitors of type **listen**:

```
pmadm -l -t listen
```

Print the per-service configuration script associated with the service **svctag** available through port monitor **pmtag**:

```
pmadm -g -p pmtag -s svctag
```

EXIT CODES

If successful, **pmadm** will exit with a status of **0**. If it fails for any reason, it will exit with a non-zero status.

FILES

```
/etc/saf/pmtag/_config  
/etc/saf/pmtag/svctag  
/var/saf/pmtag/*
```

SEE ALSO

sac(1M), **sacadm(1M)**, **doconfig(3N)**

NAME	ports – create /dev entries and inittab entries for serial lines
SYNOPSIS	<code>/usr/sbin/ports [-r <i>rootdir</i>]</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>The ports command creates symbolic links from /dev/term and /dev/cua to the serial-port character device files in /devices and adds new entries in /etc/inittab for non-system ports found. System-board ports are given single lower-case letters for names (such as a and b) while other ports are named numerically.</p> <p>ports searches the kernel device tree to find the serial devices attached to the system. It also checks /dev/term and /dev/cua to see what symbolic links to serial devices already exist. ports then performs the following:</p> <ol style="list-style-type: none"> 1. Assigns new numbers (or letters for system-board ports) to ports that are attached to the system but do not have /dev/term and /dev/cua entries. The numbers or letters assigned are the lowest-unused numbers or letters. 2. Removes dangling links: links from /dev/term and /dev/cua pointing to no-longer-existing ports. 3. Creates new /dev/term and /dev/cua links for new serial devices. 4. Invokes sacadm(1M) to make new port monitor entries for the new devices. This is not done automatically for on-board ports; on workstations these ports are often not used for dial-in sessions, so a port-monitor for one of these ports must be created explicitly. <p>If the configuration has not changed, ports exits without doing anything.</p> <p>Of interest to device driver writers are the devices that ports considers serial devices. Device nodes of type DDI_NT_SERIAL, DDI_NT_SERIAL_MB, DDI_NT_SERIAL_DO, and DDI_NT_SERIAL_MB_DO are considered serial devices (see ddi_create_minor_node(9F) for more information on node types).</p>
OPTIONS	<p><code>-r <i>rootdir</i></code> Cause ports to presume that the /dev/term, /dev/cua, and /devices directories are found under <i>rootdir</i>, not directly under /. If this argument is specified, sacadm(1M) is not invoked, since it would update terminal administration files under /etc without regard to the <i>rootdir</i>.</p>
FILES	<p>/dev/term/<i>n</i> /dev/cua/<i>n</i> /etc/inittab /etc/saf/*</p>
SEE ALSO	<p>devlinks(1M), disks(1M), pmadm(1M), sacadm(1M), tapes(1M), ddi_create_minor_node(9F)</p>

NAME	praudit – print contents of an audit trail file
SYNOPSIS	praudit [-lrs] [-ddel] [<i>filename</i> ...]
AVAILABILITY	The functionality described in this man page is available only if the Basic Security Module (BSM) has been enabled. See bsmconv(1M) for more information.
DESCRIPTION	praudit reads the listed <i>filenames</i> (or standard input, if no <i>filename</i> is specified) and interprets the data as audit trail records as defined in audit.log(4) . By default, times, user and group IDs (UIDs and GIDs, respectively) are converted to their ASCII representation. Record type and event fields are converted to their ASCII representation. A maximum of 100 audit files can be specified on the command line.
OPTIONS	-l Prints one line per record. The record type and event fields are always converted to their short ASCII representation as is done for the -s option. -r Print records in their raw form. Times, UIDs, GIDs, record types, and events are displayed as integers. This option and the -s option are exclusive. If both are used, a format usage error message is output. -s Print records in their short form. All numeric fields are converted to ASCII and displayed. The short ASCII representations for the record type and event fields are used. This option and the -r option are exclusive. If both are used, a format usage error message is output. -ddel Use <i>del</i> as the field delimiter instead of the default delimiter, which is the comma. If <i>del</i> has special meaning for the shell, it must be quoted. The maximum size of a delimiter is four characters.
FILES	/etc/security/audit_event /etc/security/audit_class
SEE ALSO	bsmconv(1M) , audit(2) , getauditflags(3) , audit.log(4) , audit_class(4) , audit_event(4) , group(4) , passwd(4)

NAME	prtconf – print system configuration
SPARC SYNOPSIS	/usr/sbin/prtconf [-F] [-p] [-P] [-v] [-V]
x86 SYNOPSIS	/usr/sbin/prtconf [-P] [-v]
AVAILABILITY	SUNWcsu
DESCRIPTION	The prtconf command prints the system configuration information. The output includes the total amount of memory, and the configuration of system peripherals formatted as a device tree.
OPTIONS	<p>-P Include information about pseudo devices. By default, information regarding pseudo devices is omitted.</p> <p>-v Specifies verbose mode.</p> <p>-F (SPARC only). Return the device pathname of the console frame buffer, if one exists. If there is no frame buffer, prtconf returns a non-zero exit code. This flag overrides all others, and returns only the name of the console, frame buffer device or a non-zero exit code. For example, if the console frame buffer on a SPARCstation 1 is cgthree in SBus slot #3, the command returns: /sbus@1,f8000000/cgthree@3,0. This option could be used to create a symlink for /dev/fb to the actual console device.</p> <p>-p (SPARC only). Displays information derived from the device tree provided by the firmware (PROM).</p> <p>-V (SPARC only). Display platform-dependent PROM version information. This flag must be used by itself, as it overrides all others. The output is a string. The format of the string is arbitrary and platform-dependent.</p>
EXIT STATUS	If successful, prtconf returns 0 . If an error occurs, prtconf prints an error message and returns 1 . For example, when an illegal option is specified, prtconf returns 1 . On a SPARC system, when the -F option is specified and the console output device is not a framebuffer, prtconf returns 1 .
SPARC EXAMPLES	<p>Running prtconf on a Sun4/65 series machine produces the following sample output:</p> <pre> example% prtconf System Configuration: Sun Microsystems sun4c Memory size: 16 Megabytes System Peripherals (Software Nodes): Sun 4_65 options, instance #0 zs, instance #0 zs, instance #1 fd (driver not attached) </pre>

```

audio (driver not attached)
sbus, instance #0
  dma, instance #0
  esp, instance #0
    sd (driver not attached)
    st (driver not attached)
    sd, instance #0
    sd, instance #1 (driver not attached)
    sd, instance #2 (driver not attached)
    sd, instance #3
    sd, instance #4 (driver not attached)
    sd, instance #5 (driver not attached)
    sd, instance #6 (driver not attached)
  le, instance #0
  cgsix (driver not attached)
auxiliary-io (driver not attached)
interrupt-enable (driver not attached)
memory-error (driver not attached)
counter-timer (driver not attached)
eeprom (driver not attached)
pseudo, instance #0

```

x86 EXAMPLES

Running **prtconf** on an x86 machine produces the following sample output:

```

example% prtconf
System Configuration: Sun Microsystems i86pc
Memory size: 32 Megabytes
System Peripherals (Software Nodes):

i86pc
  eisa, instance #0
    kd, instance #0
    ata, instance #0
      cmdk, instance #0
    aha, instance #0
      cmdk, instance #1 (driver not attached)
      cmdk, instance #2 (driver not attached)
      cmdk, instance #3 (driver not attached)
      cmdk, instance #4 (driver not attached)
      cmdk, instance #5 (driver not attached)
      cmdk, instance #6 (driver not attached)
      cmdk, instance #7
    chanmux, instance #0
    asy, instance #0
    asy, instance #1
    elx, instance #0

```

elx, instance #1 (driver not attached)
elx, instance #2 (driver not attached)
elx, instance #3 (driver not attached)
fdc, instance #0
fd, instance #0
fd, instance #1
options, instance #0
objmgr, instance #0
pseudo, instance #0
example%

SEE ALSO
SPARC Only

modinfo(1M), sysdef(1M)
openprom(7D)

NOTES

The output of the **prtconf** command is highly dependent on the version of the PROM installed in the system. The output will be affected in potentially all circumstances. The **driver not attached** message means that no driver is currently attached to that instance of the device. In general, drivers are loaded and installed (and attached to hardware instances) on demand, and when needed, and may be uninstalled and unloaded when the device is not in use.

NAME prtdiag – print system diagnostic information

SYNOPSIS /usr/platform/sun4d/sbin/prtdiag [-v]

AVAILABILITY SUNWkvm.d

DESCRIPTION **prtdiag** displays system configuration and diagnostic information. The diagnostic information lists any failed Field Replaceable Units (FRUs) in the system. **prtdiag** is supported only on sun4d machines. The interface, output, and location in the directory hierarchy for **prtdiag** are uncommitted and subject to change in future releases.

OPTIONS The following option is supported:

-v verbose mode.
 Displays the time of the most recent AC Power failure, and the most recent system watchdog information.
 This information is useful only to depot repair and manufacturing for detailed diagnostics of FRUs.

EXAMPLES The example below displays sample output from a SPARCcenter 2000 machine.

example% /usr/platform/sun4d/sbin/prtdiag

System Configuration: Sun Microsystems sun4d SPARCcenter 2000

System clock frequency: 40 MHz

Memory size: 448Mb

Number of XDBuses: 2

	CPU Units: Frequency Cache-Size				Memory Units: Group Size			
	A: MHz	MB	B: MHz	MB	0: MB	1: MB	2: MB	3: MB
Board1:	40	1.0	40	1.0	32	0	32	0
Board4:	40	1.0	40	1.0	32	32	32	32
Board7:	40	1.0	40	1.0	32	32	32	32
Board8:					32	32	32	32

=====SBus Cards=====

Board1:	0:	dma/esp(scsi)	'SUNW,500-1902'
		lebuffer/le(network)	'SUNW,500-1902'
	1:	dma/esp(scsi)	'SUNW,500-1902'
		lebuffer/le(network)	'SUNW,500-1902'
Board4:	2:	cgsix	'SUNW,501-1672'
	3:	dma/esp(scsi)	'500-1869-01'
		lebuffer/le(network)	'500-1869-01'
	0:	<empty>	
Board4:	1:	dma/esp(scsi)	'500-1902-01'
		lebuffer/le(network)	'500-1902-01'

```

                2:  bf          'SUNW,501-1732'
                3:  bf          'SUNW,501-1732'
Board7:        0:  <empty>
                1:  <empty>
                2:  <empty>
                3:  <empty>
Board8:        0:  <empty>
                1:  <empty>
                2:  <empty>
                3:  <empty>
    
```

Failed Field Replaceable Units (FRU) in System:

```

=====
SBus Card unavailable on System Board #7
Failed Field Replaceable Unit is SBus card 0

cpu-unit unavailable on System Board #8
Failed Field Replaceable Unit is SuperSPARC Module A
    
```

The example below displays sample verbose output from a SPARCcenter 2000.

example% /usr/platform/sun4d/sbin/prtdiag -v

```

System Configuration: Sun Microsystems sun4d SPARCcenter 2000
System clock frequency: 40 MHz
Memory size: 448Mb
Number of XDBuses: 2
    
```

	CPU Units: Frequency Cache-Size				Memory Units: Group Size			
	A: MHz	MB	B: MHz	MB	0: MB	1: MB	2: MB	3: MB
Board1:	40	1.0	40	1.0	32	0	32	0
Board4:	40	1.0	40	1.0	32	32	32	32
Board7:	40	1.0	40	1.0	32	32	32	32
Board8:					32	32	32	32

```

=====SBus Cards=====
Board1:        0:  dma/esp(scsi)      'SUNW,500-1902'
                lebuffer/le(network) 'SUNW,500-1902'
                1:  dma/esp(scsi)      'SUNW,500-1902'
                lebuffer/le(network) 'SUNW,500-1902'
                2:  cgsix              'SUNW,501-1672'
                3:  dma/esp(scsi)      '500-1869-01'
                lebuffer/le(network) '500-1869-01'
Board4:        0:  <empty>
                1:  dma/esp(scsi)      '500-1902-01'
                lebuffer/le(network) '500-1902-01'
                2:  bf                  'SUNW,501-1732'
                3:  bf                  'SUNW,501-1732'
Board7:        0:  <empty>
                1:  <empty>
    
```

```

                2: <empty>
                3: <empty>
Board8:        0: <empty>
                1: <empty>
                2: <empty>
                3: <empty>

```

Failed Field Replaceable Units (FRU) in System:

```

=====
SBus Card unavailable on System Board #7
Failed Field Replaceable Unit is SBus card 0

cpu-unit unavailable on System Board #8
Failed Field Replaceable Unit is SuperSPARC Module A

```

Most recent AC Power Failure:

```

=====
Sat May 22 14:21:18 1993

```

Analysis of most recent System Watchdog:

```

=====
Log Date: Thu Feb 18 22:28:15 1993

```

Analysis for Board 7

MXCC

Asynchronous Error

Error Valid, CCOP=130 ERR= 2 PA=9.10081000

BW0 (CPU B)

Client Device Error, Internal Error(s) = IOWSCE

EXIT STATUS

prtdiag returns 1 if failed FRUs are detected in the system; otherwise, it returns 0.

SEE ALSO

prtconf(1M), **psrinfo(1M)**, **modinfo(1M)**, **sysdef(1M)**, **openprom(7D)**

NAME	prtvtoc – report information about a disk geometry and partitioning
SYNOPSIS	prtvtoc [-fhs] [-t vfstab] [-m mnttab] <i>device</i>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>The prtvtoc command allows the contents of the VTOC (volume table of contents) to be viewed. The command can be used only by the super-user.</p> <p>The <i>device</i> name can be the file name of a raw device in the form of /dev/rdisk/c?t?d?s2 or can be the file name of a block device in the form of /dev/dsk/c?t?d?s2.</p>
OPTIONS	<p>The following options are supported:</p> <p>-f Report on the disk free space, including the starting block address of the free space, number of blocks, and unused partitions.</p> <p>-h Omit the headers from the normal output.</p> <p>-s Omit all headers but the column header from the normal output.</p> <p>-t vfstab Use <i>vfstab</i> as the list of filesystem defaults, in place of /etc/vfstab.</p> <p>-m mnttab Use <i>mnttab</i> as the list of mounted filesystems, in place of /etc/mnttab.</p>
EXAMPLES	<p>The command line entry and system response shown below are for a 424-megabyte hard disk:</p> <pre>example# prtvtoc /dev/rdisk/c0t3d0s2 * /dev/rdisk/c0t3d0s2 partition map * * Dimension: * 512 bytes/sector * 80 sectors/track * 9 tracks/cylinder * 720 sectors/cylinder * 2500 cylinders * 1151 accessible cylinders * * Flags: * 1: unmountable * 10: read-only *</pre>

```

*
* Partition  Tag  Flags  First Sector  Last Sector  Mount Directory
  0          2    00      0    76320    76319    /
  1          3    01    76320  132480  208799
  2          5    00      0    828720  828719
  5          6    00   208800  131760  340559  /opt
  6          4    00   340560  447120  787679  /usr
  7          8    00   787680  41040   828719  /export/home

```

example#

Codes for TAG are:

NAME	NUMBER
UNASSIGNED	0
BOOT	1
ROOT	2
SWAP	3
USR	4
BACKUP	5
VAR	7
HOME	8

FLAG indicates how the partition is to be mounted.

NAME	NUMBER
MOUNTABLE, READ AND WRITE	00
NOT MOUNTABLE	01
MOUNTABLE, READ ONLY	10

The following example shows output for the -f option for the same disk as above.

```

example# prtvtoc -f /dev/rdisk/c0t3d0s0
FREE_START=0 FREE_SIZE=0 FREE_COUNT=0 FREE_PART=4
example#

```

SEE ALSO devinfo(1M), fmthard(1M), format(1M)

WARNINGS The **mount** command does not check the "not mountable" bit.

NAME	psradm – set processors online or offline
SYNOPSIS	psradm -f -n [-v] <i>processor_id</i> psradm -a -f -n [-v]
AVAILABILITY	SUNWcsu
DESCRIPTION	psradm takes a processor offline or brings it online. An offline processor will do little or no work. The actual effect of being offline may vary from machine to machine.
OPTIONS	-f Take the specified processors offline. -n Bring the specified processors online. -a Perform the action on all processors, or as many as possible. -v Output a message giving the results of each attempted operation.
USAGE	This command is restricted to the superuser. A processor may not be taken offline if there are LWPs which are bound to the processor. On some architectures, it might not be possible to take certain processors offline if, for example, the system depends on some resource provided by the processor. At least one processor must remain online.
EXAMPLES	Set processors 2 and 3 offline: psradm -f 2 3 Set all processors online: psradm -n -a
FILES	<i>/etc/wtmp</i> for records logging processor status changes
SEE ALSO	psrinfo(1M) , p_online(2)
DIAGNOSTICS	psradm: processor 4: Invalid argument The specified processor doesn't exist in the configuration. psradm: processor 3: Device busy The specified processor could not be taken offline because it either has LWPs bound to it, is the last online processor in the system, or is needed by the system because it provides some essential service. psradm: processor 0: Not owner The user does not have permission to change processor status.

NAME	psrinfo – print processor information
SYNOPSIS	psrinfo [-v] [<i>processor_id ...</i>] psrinfo -s <i>processor_id</i>
AVAILABILITY	SUNWcsu
DESCRIPTION	psrinfo shows information on configured processors. Without arguments, it prints a line for each processor, telling whether it is online or offline, and when that status last changed.
OPTIONS	-v Use verbose mode, showing information on processor type and clock speed. -s Use silent mode. Output only 1 or 0 for the specified processor.
USAGE	Displays information about the specified processors, or all processors, if none are specified. Without arguments, information on online/offline status and time of last change is given. With the -v argument, information on processor type, floating point unit type, and clock speed are printed as well. If any of this information cannot be determined, it will be printed as “unknown.” The -s argument is provided for convenient use in shell scripts. When -s is specified, psrinfo will output 1 if the specified processor is online, and 0 otherwise.
EXAMPLES	To print the verbose status of all processors: psrinfo -v Shell usage to determine if a processor is online: <pre>if ["psrinfo -s 3 2> /dev/null" -eq 1] then echo "processor 3 is up" fi</pre>
FILES	/etc/wtmp records are read to determine last status change time.
SEE ALSO	psradm(1M) , p_online(2) , processor_info(2)
DIAGNOSTICS	psrinfo: processor 9: Invalid argument The specified processor does not exist.

NAME	putdev – edits device table
SYNOPSIS	<p>putdev <i>-a alias [attribute=value [...]]</i></p> <p>putdev <i>-m device attribute=value [attribute=value [...]]</i></p> <p>putdev <i>-d device [attribute [...]]</i></p>
DESCRIPTION	<p>putdev can add a new device to the device table, modify an existing device description or remove a device entry from the table. The first synopsis is used to add a device. The second synopsis is used to modify existing entries by adding or changing attributes. If a specified attribute is not defined, this option adds that attribute to the device definition. If a specified attribute is already defined, it modifies the attribute definition. The third synopsis is used to delete either an entire device entry or, if the attribute argument is used, to delete an attribute assignment for a device.</p>
OPTIONS	<p>-a Add a device to the device table using the specified attributes. The device must be referenced by its <i>alias</i>.</p> <p>-m Modify a device entry in the device table. If an entry already exists, it adds any specified attributes that are not defined. It also modifies any attributes which already have a value with the value specified by this command.</p> <p>-d Remove a device from the device table, when executed without the <i>attributes</i> argument. Used with the <i>attribute</i> argument, it deletes the given attribute specification for <i>device</i> from the table.</p> <p><i>alias</i> Designate the alias of the device to be added.</p> <p><i>device</i> Designate the pathname or alias of the device whose attribute is to be added, modified, or removed.</p> <p><i>attribute</i> Designate a device attribute to be added, modified, or deleted. Can be any of the device attributes described under DEVICE ATTRIBUTES except alias. This prevents an accidental modification or deletion of a device's alias from the table.</p> <p><i>value</i> Designate the value to be assigned to a device's attribute.</p>
DEVICE ATTRIBUTES	<p>The following list shows the standard device attributes, used by applications such as ufsdump(1M) and ufsrestore(1M), which can be defined for a device. You are not limited to this list, you can define any attribute you like.</p> <p>alias The unique name by which a device is known. No two devices in the database may share the same alias name. The name is limited in length to 14 characters and should contain only alphanumeric characters and the following special characters if they are escaped with a backslash: underscore (_), dollar sign (\$), hyphen (-), and period (.).</p>

bdevice	The pathname to the block special device node associated with the device, if any. The associated major/minor combination should be unique within the database and should match that associated with the cdevice field, if any. (It is the administrator's responsibility to ensure that these major/minor numbers are unique in the database.)
capacity	The capacity of the device or of the typical volume, if removable.
cdevice	The pathname to the character special device node associated with the device, if any. The associated major/minor combination should be unique within the database and should match that associated with the bdevice field, if any. (It is the administrator's responsibility to ensure that these major/minor numbers are unique in the database.)
cyl	Used by the command specified in the mkfscmd attribute.
desc	A description of any instance of a volume associated with this device (such as floppy diskette).
dpartlist	The list of disk partitions associated with this device. Used only if type=disk . The list should contain device aliases, each of which must have type=dpart .
dparttype	The type of disk partition represented by this device. Used only if type=dpart . It should be either fs (for file system) or dp (for data partition).
erasescmd	The command string that, when executed, erases the device.
fmtcmd	The command string that, when executed, formats the device.
fsname	The file system name on the file system administered on this partition, as supplied to the /usr/sbin/labelit command. This attribute is specified only if type=dpart and dparttype=fs .
gap	Used by the command specified in the mkfscmd attribute.
mkfscmd	The command string that, when executed, places a file system on a previously formatted device.
mountpt	The default mount point to use for the device. Used only if the device is mountable. For disk partitions where type=dpart and dparttype=fs , this attribute should specify the location where the partition is normally mounted.
nblocks	The number of blocks in the file system administered on this partition. Used only if type=dpart and dparttype=fs .
ninodes	The number of inodes in the file system administered on this partition. Used only if type=dpart and dparttype=fs .
norewind	The name of the character special device node that allows access to the serial device without rewinding when the device is closed.
pathname	Defines the pathname to an i-node describing the device (used for non-block or character device pathnames, such as directories).

type A token that represents inherent qualities of the device. Standard types include: 9-track, ctape, disk, directory, diskette, dpart, and qtape.

volname The volume name on the file system administered on this partition, as supplied to the `/usr/sbin/labelit` command. Used only if **type=dpart** and **dparttype=fs**.

volume A text string used to describe any instance of a volume associated with this device. This attribute should not be defined for devices which are not removable.

ERRORS

putdev will exit with one of the following values:

- 0** Successful completion of the task.
- 1** Command syntax incorrect, invalid option used, or internal error occurred.
- 2** Device table could not be opened for reading, or new device table could not be created.
- 3** If executed with the `-a` option, indicates that an entry in the device table with the alias *alias* already exists. If executed with the `-m` or `-d` options, indicates that no entry exists for device *device*.
- 4** Indicates that `-d` was requested and one or more of the specified attributes were not defined for the device.

FILES

`/etc/device.tab`

SEE ALSO

`devattr(1M)`, `putdgrp(1M)`, `ufsdump(1M)`, `ufsrestore(1M)`

System Administration Guide, Volume I

NAME	putdgrp – edits device group table
SYNOPSIS	putdgrp [-d] <i>dgroup</i> [<i>device ...</i>]
DESCRIPTION	<p>putdgrp modifies the device group table. It performs two kinds of modification. It can modify the table by creating a new device group or removing a device group. It can also change group definitions by adding or removing a device from the group definition.</p> <p>When the command is invoked with only a <i>dgroup</i> specification, the command adds the specified group name to the device group table if it does not already exist. If the -d option is also used with only the <i>dgroup</i> specification, the command deletes the group from the table.</p> <p>When the command is invoked with both a <i>dgroup</i> and a <i>device</i> specification, it adds the given device name(s) to the group definition. When invoked with both arguments and the -d option, the command deletes the device name(s) from the group definition.</p> <p>When the command is invoked with both a <i>dgroup</i> and a <i>device</i> specification and the device group does not exist, it creates the group and adds the specified devices to that new group.</p>
OPTIONS	<p>-d Delete the group or, if used with <i>device</i>, delete the device from a group definition.</p> <p><i>dgroup</i> Specify a device group name.</p> <p><i>device</i> Specify the pathname or alias of the device that is to be added to, or deleted from, the device group.</p>
ERRORS	<p>The command will exit with one of the following values:</p> <p>0 = successful completion of the task.</p> <p>1 = command syntax incorrect, invalid option used, or internal error occurred.</p> <p>2 = device group table could not be opened for reading or a new device group table could not be created.</p> <p>3 = if executed with the -d option, indicates that an entry in the device group table for the device group <i>dgroup</i> does not exist and so cannot be deleted. Otherwise, indicates that the device group <i>dgroup</i> already exists and cannot be added.</p> <p>4 = if executed with the -d option, indicates that the device group <i>dgroup</i> does not have as members one or more of the specified devices. Otherwise, indicates that the device group <i>dgroup</i> already has one or more of the specified devices as members.</p>

EXAMPLES

To add a new device group:

```
putdgrp floppies
```

To add a device to a device group:

```
putdgrp floppies diskette2
```

To delete a device group:

```
putdgrp -d floppies
```

To delete a device from a device group:

```
putdgrp -d floppies diskette2
```

FILES

/etc/dgroup.tab

SEE ALSO

listdgrp(1M), putdev(1M)

NAME	pwck, grpck – password/group file checkers
SYNOPSIS	<i>/usr/sbin/pwck</i> [<i>filename</i>] <i>/usr/sbin/grpck</i> [<i>filename</i>]
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>pwck scans the password file and notes any inconsistencies. The checks include validation of the number of fields, login name, user ID, group ID, and whether the login directory and the program-to-use-as-shell exist. The default password file is /etc/passwd.</p> <p>grpck verifies all entries in the group file. This verification includes a check of the number of fields, group name, group ID, whether any login names belong to more than NGROUPS_MAX groups and that all login names appear in the password file. The default group file is /etc/group.</p>
FILES	<i>/etc/group</i> <i>/etc/passwd</i>
SEE ALSO	<i>getpwent(3C)</i> , <i>group(4)</i> , <i>passwd(4)</i>
DIAGNOSTICS	<p>Group entries in /etc/group with no login names are flagged.</p> <p>Group file 'filename' is empty The /etc/passwd or /etc/group file is an empty file.</p> <p>cannot open file filename: No such file or directory The /etc/passwd or /etc/group file does not exist.</p>
NOTES	If no filename argument is given, grpck checks the local group file, /etc/group , and also makes sure that all login names encountered in the checked group file are known to the system getpwent(3C) routine. This means that the login names may be supplied by a network name service.

NAME	pwconv – installs and updates <code>/etc/shadow</code> with information from <code>/etc/passwd</code>
SYNOPSIS	pwconv
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>The pwconv command creates and updates <code>/etc/shadow</code> with information from <code>/etc/passwd</code>.</p> <p>pwconv relies on a special value of 'x' in the password field of <code>/etc/passwd</code>. This value of 'x' indicates that the password for the user is already in <code>/etc/shadow</code> and should not be modified.</p> <p>If the <code>/etc/shadow</code> file does not exist, this command will create <code>/etc/shadow</code> with information from <code>/etc/passwd</code>. The command populates <code>/etc/shadow</code> with the user's login name, password, and password aging information. If password aging information does not exist in <code>/etc/passwd</code> for a given user, none will be added to <code>/etc/shadow</code>. However, the last changed information will always be updated.</p> <p>If the <code>/etc/shadow</code> file does exist, the following tasks will be performed:</p> <ul style="list-style-type: none"> Entries that are in the <code>/etc/passwd</code> file and not in the <code>/etc/shadow</code> file will be added to the <code>/etc/shadow</code> file. Entries that are in the <code>/etc/shadow</code> file and not in the <code>/etc/passwd</code> file will be removed from <code>/etc/shadow</code>. Password attributes (for example, password and aging information) that exist in an <code>/etc/passwd</code> entry will be moved to the corresponding entry in <code>/etc/shadow</code>. <p>The pwconv command can only be used by the super-user.</p>
FILES	<ul style="list-style-type: none"> <code>/etc/passwd</code> <code>/etc/shadow</code> <code>/etc/passwd</code> <code>/etc/shadow</code>
SEE ALSO	passwd(1) , passmgmt(1M) , usermod(1M) , passwd(4)
DIAGNOSTICS	<p>pwconv exits with one of the following values:</p> <ul style="list-style-type: none"> 0 SUCCESS. 1 Permission denied. 2 Invalid command syntax. 3 Unexpected failure. Conversion not done. 4 Unexpected failure. Password file(s) missing. 5 Password file(s) busy. Try again later. 6 Bad entry in <code>/etc/shadow</code> file.

NAME	quot – summarize file system ownership				
SYNOPSIS	quot [-acfhnv] [<i>filesystem</i>]				
AVAILABILITY	SUNWcsu				
DESCRIPTION	quot displays the number of blocks (1024 bytes) in the named <i>filesystem</i> currently owned by each user. There is a limit of 2048 blocks. Files larger than this will be counted as a 2048 block file, but the total block count will be correct.				
OPTIONS	<p>The following options are supported:</p> <ul style="list-style-type: none"> -a Generate a report for all mounted file systems. -c Display three columns giving a file size in blocks, the number of files of that size, and a cumulative total of blocks containing files of that size or a smaller size. -f Display count of number of files as well as space owned by each user. This options is incompatible with the -c and -v options. -h Estimate the number of blocks in the file. This does not account for files with holes in them. -n Attach names to the list of files read from standard input. quot -n cannot be used alone, because it expects data from standard input. For example, the pipeline ncheck filesystem sort +0n quot -n filesystem will produce a list of all files and their owners. This option is incompatible with all other options. -v In addition to the default output, display three columns containing the number of blocks not accessed in the last 30, 60, and 90 days. 				
FILES	<table border="0" style="width: 100%;"> <tr> <td style="width: 30%;">/etc/mnttab</td> <td>mounted file systems</td> </tr> <tr> <td>/etc/passwd</td> <td>to get user names</td> </tr> </table>	/etc/mnttab	mounted file systems	/etc/passwd	to get user names
/etc/mnttab	mounted file systems				
/etc/passwd	to get user names				
SEE ALSO	du(1M)				
NOTES	This command may only be used by the super-user.				

NAME	quota – display a user’s ufs file system disk quota and usage
SYNOPSIS	quota [-v] [<i>username</i>]
DESCRIPTION	<p>quota displays users’ ufs disk usage and limits. Only the super-user may use the optional <i>username</i> argument to view the limits of other users.</p> <p>quota without options only display warnings about mounted file systems where usage is over quota. Remotely mounted file systems which do not have quotas turned on are ignored.</p> <p><i>username</i> can be the numeric UID of a user.</p>
OPTIONS	-v Display user’s quota on all mounted file systems where quotas exist.
FILES	/etc/mnttab list of currently mounted filesystems
SEE ALSO	edquota(1M), quotaon(1M), quotacheck(1M), repquota(1M), rquotad(1M)
NOTES	quota will also display quotas for NFS mounted ufs-based file systems if the rquotad daemon is running. See rquotad(1M) .

NAME	quotacheck – ufs file system quota consistency checker
SYNOPSIS	quotacheck [-p] [-v] <i>filesystem</i> . . . quotacheck -a [-pv]
DESCRIPTION	<p>quotacheck examines each mounted ufs file system, builds a table of current disk usage, and compares this table against the information stored in the file system's disk quota file. If any inconsistencies are detected, both the quota file and the current system copy of the incorrect quotas are updated.</p> <p><i>filesystem</i> is either a file system mount point or the block device on which the file system resides.</p> <p>quotacheck expects each file system to be checked to have a quota file named quotas in the root directory. If none is present, quotacheck will not check the file system.</p> <p>quotacheck accesses the character special device in calculating the actual disk usage for each user. Thus, the file systems that are checked should be quiescent while quotacheck is running.</p>
OPTIONS	<p>-p Check quotas of file systems in parallel.</p> <p>-v Indicate the calculated disk quotas for each user on a particular file system. quotacheck normally reports only those quotas modified.</p> <p>-a Check the file systems which /etc/mnttab indicates are ufs file systems. These file systems must be read-write mounted with disk quotas enabled, and have an rq entry in the mntopts field in /etc/vfstab.</p>
FILES	<p>/etc/mnttab mounted file systems</p> <p>/etc/vfstab list of default parameters for each file system</p>
SEE ALSO	edquota(1M) , quota(1M) , quotaon(1M) , repquota(1M) , quotactl(7I)

NAME	quotaon, quotaoff – turn ufs file system quotas on and off
SYNOPSIS	quotaon [-v] <i>filesystem</i> . . . quotaon -a [-v] quotaoff [-v] <i>filesystem</i> . . . quotaoff -a [-v]
DESCRIPTION	quotaon turns on disk quotas for one or more ufs file systems. Before a file system may have quotas enabled, a file named quotas , owned by root, must exist in the root directory of the file system. See edquota (1M) for details on how to modify the contents of this file. quotaoff turns off disk quotas for one or more ufs file systems. The file systems specified must already be mounted. These commands update the mntopts field of the appropriate entries in /etc/mnttab to indicate when quotas are on or off for each file system. If quotas are on, “quota” will be added to mntopts ; if quotas are off, mntopts will be marked “noquota”. <i>filesystem</i> must be either the mount point of a file system, or the block device on which the file system resides.
OPTIONS	
quotaon	-a This option is normally used at boot time to enable quotas. It applies only to those file systems in /etc/vfstab which have “rq” in the mntopts field, are currently mounted “rw”, and have a quotas file in the root directory. -v Display a message for each file system after quotas are turned on.
quotaoff	-a Force all file systems in /etc/mnttab to have their quotas disabled. -v Display a message for each file system affected.
FILES	/etc/mnttab mounted file systems /etc/vfstab list of default parameters for each file system
SEE ALSO	edquota (1M), quota (1M), quotacheck (1M), repquota (1M), mnttab (4), vfstab (4), quotactl (7l)

NAME rdate – set system date from a remote host

SYNOPSIS rdate *hostname*

DESCRIPTION rdate sets the local date and time from the *hostname* given as an argument. You must be super-user on the local system. Typically rdate can be inserted as part of a startup script.

NAME	re-preinstall – installs the JumpStart software on a system
SYNOPSIS	<i>cdrom-mnt-pt/export/exec/arch.release/sbin/install.d/re-preinstall</i> [-m <i>cdrom-mnt-pt</i>] [-k <i>platform_name</i>] <i>target-slice</i>
AVAILABILITY	SUNWcdrom (Solaris CD).
DESCRIPTION	<p>re-preinstall installs the JumpStart software (Preinstall Boot Image) on a system, so you can power-on the system and have it automatically install the Solaris software (perform a JumpStart installation on the system). Some new systems have the JumpStart software already preinstalled.</p> <p>There are two ways to use the re-preinstall command. The most common way is to run re-preinstall on a system to install the JumpStart software on its own default boot disk. This is useful if you want to restore a system to its original factory conditions.</p> <p>You can also run re-preinstall on a system to install JumpStart software on any attached disk. Once you install the JumpStart software on a disk, you can move the disk to a different system and perform a JumpStart installation on the different system.</p> <p>NOTE: While the command is valid for x86 systems, it is not useful because you cannot reboot with arguments.</p> <p>re-preinstall creates a standard file system on the specified <i>target-slice</i> (usually slice 0), and re-preinstall makes sure there is enough space on the <i>target-slice</i> for the JumpStart software, which usually requires 26 Mbytes. If sufficient space is not available, re-preinstall fails with the following message:</p> <p style="padding-left: 40px;">re-preinstall: target-slice is less than 26 Megabytes, please reformat it.</p> <p>You can use the format(1M) command to create sufficient space on the <i>target-slice</i> for the JumpStart software.</p>
OPTIONS	<p>The following options are supported:</p> <p>-m <i>cdrom-mnt-pt</i> Absolute path, or mount point, to the Solaris CD. The default is /cdrom.</p> <p>-k <i>platform_name</i> Platform name of the system that will use the disk with the JumpStart software. The default is the platform name of the system running re-preinstall. (Use the uname -i command to determine a system's platform name.)</p>
OPERANDS	<p>The following operand is supported:</p> <p><i>target-slice</i> Device name of the disk slice where the JumpStart software will be installed (usually slice 0). For example, c0t3d0s0.</p>
EXAMPLES	<p>re-preinstall on a system and install the JumpStart software on its own default boot disk:</p> <ol style="list-style-type: none"> 1. Boot the system from the Solaris CD in single-user mode at the "ok" prompt. ok boot cdrom -s

2. When the system is in single-user mode, use the **eeprom**(1M) command to determine the system's default boot disk. If no slice is specified for the system's default boot disk, the system boots from slice 0 of the specified disk, by default.

Note that when you turn on a re-preinstalled system, the system looks for the JumpStart software on the disk slice that it boots from.

3. Run the **re-preinstall** command with only the *target-slice* argument; *target-slice* must be the device name of the disk you determined in Step 2.

For example, the following command installs the JumpStart software on the system's default boot disk, **c0t3d0s0**:

```
example# /cdrom/export/exec/sparc.Solaris_2.4/sbin/install.d/re-preinstall\  
c0t3d0s0
```

EXIT STATUS

The following exit values are returned:

- 0** success
- 1** an error has occurred.

SEE ALSO

uname(1), **format(1M)**, **eeprom(1M)**, **mount(1M)**

SPARC: Installing Solaris Software

NAME	reboot – restart the operating system
SYNOPSIS	<code>/usr/sbin/reboot [-dlmq] [boot arguments]</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>reboot restarts the kernel. The kernel is loaded into memory by the PROM monitor, which transfers control to the loaded kernel.</p> <p>Although reboot can be run by the super-user at any time, shutdown(1M) is normally used first to warn all users logged in of the impending loss of service. See shutdown(1M) for details.</p> <p>reboot performs a sync(1M) operation on the disks, and then a multi-user reboot is initiated. See init(1M) for details.</p> <p>reboot normally logs the reboot to the system log daemon, syslogd(1M), and places a shutdown record in the login accounting file <code>/var/adm/wtmp</code>. These actions are inhibited if the <code>-n</code> or <code>-q</code> options are present.</p> <p>Normally, the system will reboot itself at power-up or after crashes.</p>
OPTIONS	<p><code>-d</code> Dump system core before rebooting. This option is provided for compatibility, but is not supported by the underlying reboot(3C) call.</p> <p><code>-l</code> Suppress sending a message to the system log daemon, syslogd(1M) about who executed reboot.</p> <p><code>-n</code> Avoid the sync(1M) operation. This option can be used if a disk or the processor is on fire.</p> <p><code>-q</code> Quick. Reboot quickly and ungracefully, without shutting down running processes first.</p> <p><i>boot arguments</i></p> <p>These arguments are accepted for compatibility, and are passed unchanged to the uadmin(2) system call.</p> <p>On x86 systems only, note that currently, boot arguments are not passed on to the boot program, so they have no effect. You must type in the arguments when responding to the boot prompt ">" to have the desired effect.</p>
EXAMPLES	<p>In the example below, the delimiter ‘—’ (two hyphens) must be used to separate the options of reboot from the arguments of boot(1M).</p> <pre>example# reboot -dl — -rv</pre>
FILES	<code>/var/adm/wtmp</code> login accounting file
SEE ALSO	boot(1M) , crash(1M) , fsck(1M) , halt(1M) , init(1M) , shutdown(1M) , sync(1M) , syslogd(1M) , uadmin(2) , reboot(3C)

NAME	rem_drv – remove a device driver from the system
SYNOPSIS	rem_drv [-b <i>basedir</i>] <i>device_driver</i>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>The rem_drv command informs the system that the device driver <i>device_driver</i> is no longer valid. If possible, rem_drv unloads <i>device_driver</i> from memory. Entries for the device in the /devices namespace are removed. rem_drv also updates the system driver configuration files.</p> <p>If rem_drv has been executed, the next time the system is rebooted it will automatically perform a reconfiguration boot (see kernel(1M)).</p>
OPTIONS	<p>-b <i>basedir</i> Sets the path to the root directory of the diskless client. Used on the server to execute rem_drv for a client. The client machine must be rebooted to unload the driver.</p>
EXAMPLES	<p>The following example removes the sd driver from use:</p> <pre>example% rem_drv sd</pre> <p>The next example removes the driver from the sun1 diskless client. The driver will not be uninstalled nor unloaded until the client machine is rebooted.</p> <pre>example% rem_drv -b /export/root/sun1 sd</pre>
SEE ALSO	add_drv(1M) , drvconfig(1M) , kernel(1M)

NAME	removef – remove a file from software database
SYNOPSIS	removef <i>pkginst</i> <i>path1</i> [<i>path2</i> ...] removef -f <i>pkginst</i>
AVAILABILITY	SUNWcsu
DESCRIPTION	removef informs the system that the user, or software, intends to remove a pathname. Output from removef is the list of input pathnames that may be safely removed (no other packages have a dependency on them).
OPTIONS	-f After all files have been processed, removef should be invoked with the -f option to indicate that the removal phase is complete.
EXAMPLES	The following shows the use of removef in an optional pre-install script: <pre>echo "The following files are no longer part of this package and are being removed." removef \$PKGINST /dev/xt[0-9][0-9][0-9] while read pathname do echo "\$pathname" rm -f \$pathname done removef -f \$PKGINST exit 2</pre>
SEE ALSO	pkginfo(1) , pkgmk(1) , pkgparam(1) , pkgproto(1) , pkgtrans(1) , installf(1M) , pkgadd(1M) , pkgask(1M) , pkgchk(1M)

NAME	repquota – summarize quotas for a ufs file system
SYNOPSIS	repquota [-v] <i>filesystem</i> ... repquota -a [-v]
DESCRIPTION	repquota prints a summary of the disk usage and quotas for the specified ufs file systems. The current number of files and amount of space (in kilobytes) is printed for each user along with any quotas created with edquota (1M). The <i>filesystem</i> must have the file quotas in its root directory. Only the super-user may view quotas which are not their own.
OPTIONS	-a Report on all mounted ufs file systems that have rq in the mntopts field of the /etc/vfstab file. -v Report quotas for all users, even those who do not consume resources.
SEE ALSO	edquota (1M), quota (1M), quotacheck (1M), quotaon (1M), quotactl (7I)

NAME	rmmount – removable media mounter that automatically mounts a file system on a CD-ROM and floppy																
SYNOPSIS	<code>/usr/sbin/rmmount [-D]</code>																
DESCRIPTION	<p>rmmount is a removable media mounter that is executed by Volume Management whenever a CD-ROM or floppy is inserted. The Volume Management daemon, vold(1M), manages CD-ROM and floppy devices.</p> <p>Upon insertion, rmmount determines what type of file system (if any) is on the media. If a file system is present, rmmount mounts the file system in one of the following locations.</p> <table border="0"> <thead> <tr> <th style="text-align: left;">Mount Location</th> <th style="text-align: left;">State of Media</th> </tr> </thead> <tbody> <tr> <td><code>/floppy/floppy0</code></td> <td>symbolic link to mounted floppy in local floppy drive</td> </tr> <tr> <td><code>/floppy/floppy_name</code></td> <td>mounted named floppy</td> </tr> <tr> <td><code>/floppy/unnamed_floppy</code></td> <td>mounted unnamed floppy</td> </tr> <tr> <td><code>/cdrom/cdrom0</code></td> <td>symbolic link to mounted CD-ROM in local CD-ROM drive</td> </tr> <tr> <td><code>/cdrom/CD-ROM_name</code></td> <td>mounted named CD-ROM</td> </tr> <tr> <td><code>/cdrom/CD-ROM_name/partition</code></td> <td>mounted named CD-ROM with partitioned file system</td> </tr> <tr> <td><code>/cdrom/unnamed_cdrom</code></td> <td>mounted unnamed CD-ROM</td> </tr> </tbody> </table> <p>If the media is read-only (either CD-ROM or floppy with write-protect tab set), the file system is mounted read-only.</p> <p>If a file system is not identified, rmmount does not mount a file system. See the <i>System Administration Guide, Volume I</i> for more information on the location of CD-ROM and floppy media without file systems. Also see volfs(7FS).</p> <p>If a file system type has been determined, it is then checked to see that it is “clean.” If the file system is “dirty,” fsck -p (see fsck(1M)) is run in an attempt to clean it. If fsck fails, the file system is mounted read-only.</p> <p>After the mount is complete, “actions” associated with the media type are executed. These actions allow for the notification to other programs that new media are available. These actions are shared objects and are described in the configuration file, <code>/etc/rmmount.conf</code>.</p> <p>Actions are executed in the order in which they appear in the configuration file. The action function can return either 1 or 0. If it returns 0, no further actions will be executed. This allows the function to control which applications are executed.</p> <p>In order to execute an action, rmmount performs a dlopen(3X) on the shared object and calls the action function defined within it. The definition of the interface to actions can be found in <code>/usr/include/rmmount.h</code>.</p>	Mount Location	State of Media	<code>/floppy/floppy0</code>	symbolic link to mounted floppy in local floppy drive	<code>/floppy/floppy_name</code>	mounted named floppy	<code>/floppy/unnamed_floppy</code>	mounted unnamed floppy	<code>/cdrom/cdrom0</code>	symbolic link to mounted CD-ROM in local CD-ROM drive	<code>/cdrom/CD-ROM_name</code>	mounted named CD-ROM	<code>/cdrom/CD-ROM_name/partition</code>	mounted named CD-ROM with partitioned file system	<code>/cdrom/unnamed_cdrom</code>	mounted unnamed CD-ROM
Mount Location	State of Media																
<code>/floppy/floppy0</code>	symbolic link to mounted floppy in local floppy drive																
<code>/floppy/floppy_name</code>	mounted named floppy																
<code>/floppy/unnamed_floppy</code>	mounted unnamed floppy																
<code>/cdrom/cdrom0</code>	symbolic link to mounted CD-ROM in local CD-ROM drive																
<code>/cdrom/CD-ROM_name</code>	mounted named CD-ROM																
<code>/cdrom/CD-ROM_name/partition</code>	mounted named CD-ROM with partitioned file system																
<code>/cdrom/unnamed_cdrom</code>	mounted unnamed CD-ROM																

File systems mounted by **rmmount** are always mounted with the **nosuid** flag set, thereby disabling set-uid programs and access to block or character devices in that file system. Upon ejection, **rmmount** unmounts mounted file systems and executes actions associated with the media type. If a file system is “busy” (i.e., it contains the current working directory of a live process), the ejection will fail.

OPTIONS

-D Turn on the debugging output from the **rmmount** dprintf calls.

FILES

/etc/rmmount.conf removable media mounter configuration file.
/usr/lib/rmmount/*.so.1 shared objects used by **rmmount**.

SEE ALSO

volcancel(1), **volcheck(1)**, **volmissing(1)** **fsck(1M)**, **vold(1M)**, **dlopen(3X)**,
rmmount.conf(4), **vold.conf(4)**, **volfs(7FS)**,
System Administration Guide, Volume I

NAME	rmt – remote magtape protocol module
SYNOPSIS	<code>/usr/sbin/rmt</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>rmt is a program used by the remote dump and restore programs in manipulating a magnetic tape drive through an interprocess communication connection. rmt is normally started up with an rexec(3N) or rcmd(3N) call.</p> <p>The rmt program accepts requests that are specific to the manipulation of magnetic tapes, performs the commands, then responds with a status indication. All responses are in ASCII and in one of two forms. Successful commands have responses of</p> <p style="padding-left: 40px;"><i>A</i>number\n</p> <p>where <i>number</i> is an ASCII representation of a decimal number. Unsuccessful commands are responded to with</p> <p style="padding-left: 40px;"><i>Error-number</i>\n<i>error-message</i>\n</p> <p>where <i>error-number</i> is one of the possible error numbers described in intro(2), and <i>error-message</i> is the corresponding error string as printed from a call to perror(3C). The protocol consists of the following commands:</p> <p style="padding-left: 40px;">S\n Return the status of the open device, as obtained with a MTIOCGET ioctl call. If the operation was successful, an “ack” is sent with the size of the status buffer, then the status buffer is sent (in binary).</p> <p style="padding-left: 40px;">C<i>device</i>\n Close the currently open device. The <i>device</i> specified is ignored.</p> <p style="padding-left: 40px;">I<i>operation</i>\n<i>count</i>\n Perform a MTIOCOP ioctl(2) command using the specified parameters. The parameters are interpreted as the ASCII representations of the decimal values to place in the <i>mt_op</i> and <i>mt_count</i> fields of the structure used in the ioctl call. When the operation is successful the return value is the <i>count</i> parameter.</p> <p style="padding-left: 40px;">L<i>whence</i>\n<i>offset</i>\n Perform an lseek(2) operation using the specified parameters. The response value is returned from the lseek call.</p> <p style="padding-left: 40px;">O<i>device</i>\n<i>mode</i>\n Open the specified <i>device</i> using the indicated <i>mode</i>. <i>device</i> is a full pathname, and <i>mode</i> is an ASCII representation of a decimal number suitable for passing to open(9E). If a device is already open, it is closed before a new open is performed.</p>

Rcount\n Read *count* bytes of data from the open device. **rmt** performs the requested **read(9E)** and responds with **Acount-read\n** if the read was successful; otherwise an error in standard format is returned. If the read was successful, the data read is sent.

Wcount\n Write data onto the open device. **rmt** reads *count* bytes from the connection, aborting if a premature EOF is encountered. The response value is returned from the **write(9E)** call.

Any other command causes **rmt** to exit.

SEE ALSO **ufsdump(1M)**, **ufsrestore(1M)**, **intro(2)**, **ioctl(2)**, **lseek(2)**, **perror(3C)**, **rcmd(3N)**, **rexec(3N)**, **mtio(7I)**, **open(9E)**, **read(9E)**, **write(9E)**

DIAGNOSTICS All responses are of the form described above.

BUGS Do not use this for a remote file access protocol.

NAME	route – manually manipulate the routing tables
SYNOPSIS	route [-fn] add delete [host net] <i>destination</i> [<i>gateway</i> [<i>metric</i>]]
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>route manually manipulates the network routing tables normally maintained by the system routing daemon, routed(1M), or through default routes and redirect messages from routers. route allows the super-user to operate directly on the routing table for the specific host or network indicated by <i>destination</i>. default is available for gateways to use after all other routes have been attempted. The <i>gateway</i> argument, if present, indicates the network gateway to which packets should be addressed. The <i>metric</i> argument indicates the number of “hops” to the <i>destination</i>. The <i>metric</i> is required for add commands; it must be 0 if the destination is on a directly attached network, and non-zero if the route utilizes one or more gateways.</p> <p>A <i>metric</i> of 0 implies that the route does not refer to a gateway, but rather to one of the machine’s interfaces. Destinations matching such a route are sent out on the interface identified by the <i>gateway</i> address. For interfaces using the ARP protocol, a <i>metric</i> of 0 is used to specify <i>all destinations are local</i>, meaning that a host should ARP for all addresses by adding a default route containing a <i>metric</i> of 0, as illustrated in the following example:</p> <pre style="margin-left: 40px;">route add default <hostname> 0</pre> <p><i><hostname></i> is the name or IP address associated with the network interface all packets should be sent over. On a host with a single network interface, <i><hostname></i> is normally the same as the <i><nodename></i> returned by uname -n.</p> <p>The add command instructs route to add a route to <i>destination</i>. delete deletes a route. Routes to a particular host must be distinguished from those to a network. The optional keywords net and host force the destination to be interpreted as a network or a host, respectively. Otherwise, if the destination has a “local address part” of INADDR_ANY, then the route is assumed to be to a network; otherwise, it is presumed to be a route to a host. If the route is to a destination connected by a gateway, the <i>metric</i> parameter should be greater than 0. If adding a route with <i>metric</i> 0, the gateway given is the address of this host on the common network, indicating the interface to be used directly for transmission. All symbolic names specified for a <i>destination</i> (except default) or <i>gateway</i> are looked up in the hosts database using gethostbyname(3N). If this lookup fails, then the name is looked up in the networks database using getnetbyname(3N). default is also a valid destination, which is used for all routes if there is no specific host or network route.</p>
OPTIONS	<p>-f Flush the routing tables of all gateway entries. If this is used in conjunction with one of the commands described above, route flushes the gateways before performing the command.</p> <p>-n Prevent attempts to print host and network names symbolically when reporting actions. This is useful, for example, when all name servers are down on your local net, and you need a route before you can contact the name server.</p>

NOTES	<p><i>All destinations are local</i> assumes that the routers implement proxy arp. Normally, using router discovery (see in.rdisc(1M)) is more reliable than using proxy arp.</p> <p>Combining the <i>all destinations are local</i> route with subnet or network routes can lead to unpredictable results: the search order as it relates to the <i>all destinations are local</i> route are undefined and may vary from release to release.</p>
FILES	<p>/etc/hosts /etc/networks</p>
SEE ALSO	<p>netstat(1M), routed(1M), ioctl(2), gethostbyname(3N), getnetbyname(3N), routing(4)</p>
DIAGNOSTICS	<p>add [host net] destination:gateway The specified route is being added to the tables. The values printed are from the routing table entry supplied in the ioctl(2) call.</p> <p>delete [host net] destination:gateway The specified route is being deleted.</p> <p>destination done When the -f flag is specified, each routing table entry deleted is indicated with a message of this form.</p> <p>Network is unreachable An attempt to add a route failed because the gateway listed was not on a directly-connected network. Give the next-hop gateway instead.</p> <p>not in table A delete operation was attempted for an entry that is not in the table.</p> <p>routing table overflow An add operation was attempted, but the system was unable to allocate memory to create the new entry.</p>

NAME	rpc.bootparamd, bootparamd – boot parameter server
SYNOPSIS	<code>/usr/sbin/rpc.bootparamd [-d]</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>rpc.bootparamd is a server process that provides information from a bootparams database to diskless clients at boot time. See bootparams(4)</p> <p>The source for the bootparams database is determined by the nsswitch.conf(4) file (on the machine running the rpc.bootparamd process).</p> <p>The rpc.bootparamd program can be invoked either by inetd(1M) or directly from the command line.</p>
OPTIONS	-d Display debugging information.
FILES	<code>/etc/bootparams</code> boot parameter data base <code>/etc/nsswitch.conf</code> configuration file for the name-service switch
SEE ALSO	inetd(1M) , bootparams(4) , nsswitch.conf(4)
NOTES	<p>A diskless client requires service from at least one rpc.bootparamd process running on a server that is on the same IP subnetwork as the diskless client.</p> <p>Some routines that compare hostnames use case-sensitive string comparisons; some do not. If an incoming request fails, verify that the case of the hostname in the file to be parsed matches the case of the hostname called for, and attempt the request again.</p>

NAME	rpc.nisd, nisd – NIS+ service daemon
SYNOPSIS	<code>/usr/sbin/rpc.nisd [-ACDFhlv] [-Y [-B [-t netid]]] [-d dictionary] [-L load] [-S level]</code>
AVAILABILITY	SUNWnisu
DESCRIPTION	<p>The rpc.nisd daemon is an RPC service that implements the NIS+ service. This daemon must be running on all machines which serve a portion of the NIS+ namespace.</p> <p>rpc.nisd is usually started from a system startup script.</p> <p>The -B option causes rpc.nisd to start an auxiliary process, rpc.nisd_resolv, which provides ypserv compatible DNS forwarding for NIS host requests. rpc.nisd_resolv can also be started independently. See rpc.nisd_resolv(1M) for more information on using rpc.nisd_resolv independently.</p>
OPTIONS	<p>-A Authentication verbose mode. The daemon logs all the authentication related activities to syslogd(1M) with LOG_INFO priority.</p> <p>-C Open diagnostic channel on /dev/console.</p> <p>-D Debug mode (don't fork).</p> <p>-F Force the server to do a checkpoint of the database when it starts up. Forced checkpoints may be required when the server is low on disk space. This option removes updates from the transaction log that have propagated to all of the replicas.</p> <p>-h Print list of options.</p> <p>-v Verbose. With this option, the daemon sends a running narration of what it is doing to the syslog daemon (see syslogd(1M)) at LOG_INFO priority. This option is most useful for debugging problems with the service (see also -A option).</p> <p>-Y Put the server into NIS (YP) compatibility mode. When operating in this mode, the NIS+ server will respond to NIS Version 2 requests using the version 2 protocol. Because the YP protocol is not authenticated, only those items that have read access to nobody (the unauthenticated request) will be visible through the V2 protocol. It supports only the standard Version 2 maps in this mode (see -B option and NOTES in ypfiles(4)).</p> <p>-B Provide ypserv compatible DNS forwarding for NIS host requests. The DNS resolving process, rpc.nisd_resolv, is started and controlled by rpc.nisd. This option requires that the /etc/resolv.conf file be setup for communication with a DNS nameserver. The nslookup utility can be used to verify communication with a DNS nameserver. See resolv.conf(4) and nslookup(1M).</p>

- t *netid*** Use *netid* as the transport for communication between **rpc.nisd** and **rpc.nisd_resolv**. The default transport is **ticots(7D)** (**tcp** on SunOS 4.x systems).
- d *dictionary*** Specify an alternate dictionary for the NIS+ database. The primary use of this option is for testing. Note that the string is not interpreted, rather it is simply passed to the **db_initialize** function. See **nis_db(3N)**.
- L *number*** Specify the “load” the NIS+ service is allowed to place on the server. The load is specified in terms of the *number* of child processes that the server may spawn. This *number must* be at least 1 for the callback functions to work correctly. The default is 128.
- S *level*** Set the authorization security level of the service. The argument is a number between 0 and 2. By default, the daemon runs at security level 2.
- 0** Security level 0 is designed to be used for testing and initial setup of the NIS+ namespace. When running at level 0, the daemon does not enforce any access controls. Any client is allowed to perform any operation, including updates and deletions.
 - 1** At security level 1, the daemon accepts both **AUTH_SYS** and **AUTH_DES** credentials for authenticating clients and authorizing them to perform NIS+ operations. This is not a secure mode of operation since **AUTH_SYS** credentials are easily forged. It should not be used on networks in which any untrusted users may potentially have access.
 - 2** At security level 2, the daemon accepts only **AUTH_DES** credentials for authentication and authorization. This is the highest level of security currently provided by the NIS+ service. This is the default security level if the **-S** option is not used.

EXAMPLES

The following example sets up the NIS+ service.

```
example% rpc.nisd
```

The following example sets up the NIS+ service, emulating YP with DNS forwarding.

```
example% rpc.nisd -YB
```

ENVIRONMENT

NETPATH The transports that the NIS+ service will use can be limited by setting this environment variable (see **netconfig(4)**).

FILES

/var/nis/parent.object This file contains an XDR encoded NIS+ object that describes the namespace above a root server. This parent namespace may be another NIS+ namespace or a foreign namespace such as one served by the Domain Name Service. It is only present on servers that are serving the root of the namespace.

/var/nis/root.object This file contains an XDR encoded NIS+ object that describes the root of the namespace. It is only present on servers that are serving the root of the namespace.

/etc/init.d/rpc initialization script for NIS+

SEE ALSO

nis_cachemgr(1M), **nisinit(1M)**, **nissetup(1M)**, **nslookup(1M)**, **syslogd(1M)**,
rpc.nisd_resolv(1M), **rpc.nispasswd(1M)**, **nis_db(3N)**, **netconfig(4)**, **nisfiles(4)**,
resolv.conf(4), **ypfiles(4)**, **ticots(7D)**

NAME	rpc.nisd_resolv, nisd_resolv – NIS+ service daemon
SYNOPSIS	rpc.nisd_resolv [<i>-v</i> <i>-V</i>] [<i>-F</i> [<i>-C fd</i>]] [<i>-t xx</i>] [<i>-p yy</i>]
AVAILABILITY	SUNWnisu
DESCRIPTION	rpc.nisd_resolv is an auxiliary process to rpc.nisd used to provide ypserv compatible DNS forwarding for NIS host requests. It is generally started by invoking rpc.nisd with the <i>-B</i> option. rpc.nisd_resolv can also be started independently with the following options.
OPTIONS	<i>-F</i> Run in foreground. <i>-C fd</i> Use <i>fd</i> for service xprt (from nisd). <i>-v</i> Verbose. Send output to the syslog daemon. <i>-V</i> Verbose. Send output to stdout. <i>-t xx</i> Use transport <i>xx</i> . <i>-p yy</i> Use transient program# <i>yy</i> .
SEE ALSO	nslookup (1M), rpc.nisd (1M), resolv.conf (4).
NOTES	This command requires that the /etc/resolv.conf file be setup for communication with a DNS nameserver. The nslookup utility can be used to verify communication with a DNS nameserver. See resolv.conf (4) and nslookup (1M).

NAME	rpc.nispasswd, nispasswd – NIS+ password update daemon
SYNOPSIS	<code>/usr/sbin/rpc.nispasswd [-a <i>attempts</i>] [-c <i>minutes</i>] [-D] [-g] [-v]</code>
AVAILABILITY	SUNWnisu
DESCRIPTION	<p>rpc.nispasswd daemon is an ONC+ RPC service that services password update requests from nispasswd(1) and yppasswd(1). It updates password entries in the NIS+ passwd table.</p> <p>rpc.nispasswd is normally started from a system startup script after the NIS+ server (rpc.nisd(1M)) has been started. rpc.nispasswd will determine whether it is running on a machine that is a master server for one or more NIS+ directories. If it discovers that the host is not a master server, then it will promptly exit. It will also determine if rpc.nisd(1M) is running in NIS(YP) compatibility mode (the -Y option) and will register as yppasswd for NIS(YP) clients as well.</p> <p>rpc.nispasswd will syslog all failed password update attempts, which will allow an administrator to determine whether someone was trying to "crack" the passwords.</p> <p>rpc.nispasswd has to be run by a superuser.</p>
OPTIONS	<p>-a <i>attempts</i> Set the maximum number of attempts allowed to authenticate the caller within a password update request session. Failed attempts are syslogd(1M) and the request is cached by the daemon. After the maximum number of allowed attempts the daemon severs the connection to the client. The default value is set to 3.</p> <p>-c <i>minutes</i> Set the number of minutes a failed password update request should be cached by the daemon. This is the time during which if the daemon receives further password update requests for the same user and authentication of the caller fails, then the daemon will simply not respond. The default value is set to 30 minutes.</p> <p>-D Debug. Run in debugging mode.</p> <p>-g Generate DES credential. By default the DES credential is not generated for the user if they do not have one. By specifying this option, if the user does not have a credential, then one will be generated for them and stored in the NIS+ cred table.</p> <p>-v Verbose. With this option, the daemon sends a running narration of what it is doing to the syslog daemon. This option is useful for debugging problems.</p>
EXIT STATUS	<p>0 success</p> <p>1 an error has occurred.</p>

FILES /etc/init.d/rpc initialization script for NIS+

SEE ALSO nispasswd(1), passwd(1), yppasswd(1), rpc.nisd(1M), syslogd(1M), nsswitch.conf(4)

NAME	rpc.rexd, rexd – RPC-based remote execution server						
SYNOPSIS	<code>/usr/sbin/rpc.rexd [-s]</code>						
AVAILABILITY	SUNWnisu						
DESCRIPTION	<p>rpc.rexd is the Sun RPC server for remote program execution. This daemon is started by inetd(1M) whenever a remote execution request is made.</p> <p>For non-interactive programs, the standard file descriptors are connected directly to TCP connections. Interactive programs involve pseudo-terminals, in a fashion that is similar to the login sessions provided by rlogin(1). This daemon may use NFS to mount file systems specified in the remote execution request.</p>						
OPTIONS	<p>-s Secure. When specified, requests must have valid DES credentials. If the request does not have a DES credential it is rejected. The default publickey credential is rejected. Only newer on(1) commands send DES credentials.</p> <p>If access is denied with an authentication error, you may have to set your publickey with the chkey(1) command.</p> <p>Specifying the -s option without presenting secure credentials will result in an error message: Unix too weak auth (DesONLY)!</p>						
FILES	<table border="0"> <tr> <td style="padding-right: 20px;"><code>/dev/ptsn</code></td> <td>pseudo-terminals used for interactive mode</td> </tr> <tr> <td><code>/etc/passwd</code></td> <td>authorized users</td> </tr> <tr> <td><code>/tmp_rex/rexd?????</code></td> <td>temporary mount points for remote file systems.</td> </tr> </table>	<code>/dev/ptsn</code>	pseudo-terminals used for interactive mode	<code>/etc/passwd</code>	authorized users	<code>/tmp_rex/rexd?????</code>	temporary mount points for remote file systems.
<code>/dev/ptsn</code>	pseudo-terminals used for interactive mode						
<code>/etc/passwd</code>	authorized users						
<code>/tmp_rex/rexd?????</code>	temporary mount points for remote file systems.						
SEE ALSO	chkey (1), on (1), rlogin (1), inetd (1M), inetd.conf (4), publickey (4)						
DIAGNOSTICS	Diagnostic messages are normally printed on the console, and returned to the requestor.						
NOTES	Root cannot execute commands using rexd client programs such as on (1).						

NAME	rpc.rstatd, rstatd – kernel statistics server
SYNOPSIS	/usr/lib/netsvc/rstat/rpc.rstatd
DESCRIPTION	rpc.rstatd is a server which returns performance statistics obtained from the kernel. rup(1) uses rpc.rstatd to collect the <i>uptime</i> information that it displays. rpc.rstatd is an RPC service.
SEE ALSO	rup(1) , inetd(1M) , services(4)

NAME	rpc.rusersd, rusersd – network username server
SYNOPSIS	/usr/lib/netsvc/rusers/rpc.rusersd
DESCRIPTION	rpc.rusersd is a server that returns a list of users on the host. The rpc.rusersd daemon may be started by inetd(1M) or listen(1M) .
SEE ALSO	inetd(1M) , listen(1M) , pmadm(1M) , sacadm(1M)

NAME	rpc.rwalld, rwalld – network rwall server
SYNOPSIS	/usr/lib/netsvc/rwall/rpc.rwalld
DESCRIPTION	rpc.rwalld is a server that handles rwall(1M) requests. It is implemented by calling wall(1M) on all the appropriate network machines. The rpc.rwalld daemon may be started by inetd(1M) or listen(1M) .
SEE ALSO	inetd(1M) , listen(1M) , rwall(1M) , wall(1M)

NAME	rpc.sprayd, sprayd – spray server
SYNOPSIS	/usr/lib/netsvc/spray/rpc.sprayd
DESCRIPTION	<p>rpc.sprayd is a server that records the packets sent by spray(1M). The rpc.sprayd daemon may be started by inetd(1M) or listen(1M).</p> <p>The service provided by rpc.sprayd is not useful as a networking benchmark as it uses unreliable connectionless transports, (udp for example). It can report a large number of packets dropped when the drops were caused by the program sending packets faster than they can be buffered locally (before the packets get to the network medium).</p>
SEE ALSO	inetd(1M) listen(1M) , pmadm(1M) , sacadm(1M) , spray(1M)

NAME	rpcbind – universal addresses to RPC program number mapper
SYNOPSIS	rpcbind [-d] [-w]
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>rpcbind is a server that converts RPC program numbers into universal addresses. It must be running on the host to be able to make RPC calls on a server on that machine.</p> <p>When an RPC service is started, it tells rpcbind the address at which it is listening, and the RPC program numbers it is prepared to serve. When a client wishes to make an RPC call to a given program number, it first contacts rpcbind on the server machine to determine the address where RPC requests should be sent.</p> <p>rpcbind should be started before any other RPC service. Normally, standard RPC servers are started by port monitors, so rpcbind must be started before port monitors are invoked.</p> <p>When rpcbind is started, it checks that certain name-to-address translation-calls function correctly. If they fail, the network configuration databases may be corrupt. Since RPC services cannot function correctly in this situation, rpcbind reports the condition and terminates.</p> <p>rpcbind can only be started by the super-user.</p>
OPTIONS	<p>-d Run in debug mode. In this mode, rpcbind will not fork when it starts, will print additional information during operation, and will abort on certain errors. With this option, the name-to-address translation consistency checks are shown in detail.</p> <p>-w Do a warm start. If rpcbind aborts or terminates on SIGINT or SIGTERM, it will write the current list of registered services to /tmp/portmap.file and /tmp/rpcbind.file. Starting rpcbind with the -w option instructs it to look for these files and start operation with the registrations found in them. This allows rpcbind to resume operation without requiring all RPC services to be restarted.</p>
NOTES	<p>Terminating rpcbind with SIGKILL will prevent the warm-start files from being written. All RPC servers must be restarted if the following occurs: rpcbind crashes (or is killed with SIGKILL) and is unable to write the warm-start files; rpcbind is started without the -w option after a graceful termination; or, the warm-start files are not found by rpcbind.</p>
SEE ALSO	rpcinfo (1M), rpcbind (3N)
FILES	/tmp/portmap.file /tmp/rpcbind.file

NAME	rpcinfo – report RPC information
SYNOPSIS	<pre> rpcinfo [-m] [-s] [<i>host</i>] rpcinfo -p [<i>host</i>] rpcinfo -T <i>transport</i> <i>host</i> <i>prognum</i> [<i>versnum</i>] rpcinfo -l [-T <i>transport</i>] <i>host</i> <i>prognum</i> [<i>versnum</i>] rpcinfo [-n <i>portnum</i>] -u <i>host</i> <i>prognum</i> [<i>versnum</i>] rpcinfo [-n <i>portnum</i>] -t <i>host</i> <i>prognum</i> [<i>versnum</i>] rpcinfo -a <i>serv_address</i> -T <i>transport</i> <i>prognum</i> [<i>versnum</i>] rpcinfo -b [-T <i>transport</i>] <i>prognum</i> <i>versnum</i> rpcinfo -d [-T <i>transport</i>] <i>prognum</i> <i>versnum</i> </pre>
DESCRIPTION	<p>rpcinfo makes an RPC call to an RPC server and reports what it finds.</p> <p>In the first synopsis, rpcinfo lists all the registered RPC services with rpcbind on <i>host</i>. If <i>host</i> is not specified, the local host is the default. If -s is used, the information is displayed in a concise format.</p> <p>In the second synopsis, rpcinfo lists all the RPC services registered with rpcbind, version 2. Also note that the format of the information is different in the first and the second synopsis. This is because the second synopsis is an older protocol used to collect the information displayed (version 2 of the rpcbind protocol).</p> <p>The third synopsis makes an RPC call to procedure 0 of <i>prognum</i> and <i>versnum</i> on the specified <i>host</i> and reports whether a response was received. <i>transport</i> is the transport which has to be used for contacting the given service. The remote address of the service is obtained by making a call to the remote rpcbind.</p> <p>The <i>prognum</i> argument is a number that represents an RPC program number (see rpc(4)). If a <i>versnum</i> is specified, rpcinfo attempts to call that version of the specified <i>prognum</i>. Otherwise, rpcinfo attempts to find all the registered version numbers for the specified <i>prognum</i> by calling version 0, which is presumed not to exist; if it does exist, rpcinfo attempts to obtain this information by calling an extremely high version number instead, and attempts to call each registered version. Note that the version number is required for -b and -d options.</p> <p>The other ways of using rpcinfo are described in the EXAMPLES section.</p>
OPTIONS	<p>-T <i>transport</i> Specify the transport on which the service is required. If this option is not specified, rpcinfo uses the transport specified in the NETPATH environment variable, or if that is unset or null, the transport in the netconfig(4) database is used. This is a generic option, and can be used in conjunction with other options as shown in the SYNOPSIS.</p> <p>-a <i>serv_address</i> Use <i>serv_address</i> as the (universal) address for the service on <i>transport</i> to ping procedure 0 of the specified <i>prognum</i> and report whether a response was received. The -T option is required with the -a option.</p>

- If *versnum* is not specified, **rpcinfo** tries to ping all available version numbers for that program number. This option avoids calls to remote **rpcbind** to find the address of the service. The *serv_address* is specified in universal address format of the given transport.
- b** Make an RPC broadcast to procedure **0** of the specified *prognum* and *versnum* and report all hosts that respond. If *transport* is specified, it broadcasts its request only on the specified transport. If broadcasting is not supported by any transport, an error message is printed. Use of broadcasting should be limited because of the potential for adverse effect on other systems.
 - d** Delete registration for the RPC service of the specified *prognum* and *versnum*. If *transport* is specified, unregister the service on only that transport, otherwise unregister the service on all the transports on which it was registered. Only the owner of a service can delete a registration, except the super-user who can delete any service.
 - l** Display a list of entries with a given *prognum* and *versnum* on the specified *host*. Entries are returned for all transports in the same protocol family as that used to contact the remote **rpcbind**.
 - m** Display a table of statistics of **rpcbind** operations on the given *host*. The table shows statistics for each version of **rpcbind** (versions 2, 3 and 4), giving the number of times each procedure was requested and successfully serviced, the number and type of remote call requests that were made, and information about RPC address lookups that were handled. This is useful for monitoring RPC activities on *host*.
 - n portnum** Use *portnum* as the port number for the **-t** and **-u** options instead of the port number given by **rpcbind**. Use of this option avoids a call to the remote **rpcbind** to find out the address of the service. This option is made obsolete by the **-a** option.
 - p** Probe **rpcbind** on *host* using version 2 of the **rpcbind** protocol, and display a list of all registered RPC programs. If *host* is not specified, it defaults to the local host. Note that version 2 of the **rpcbind** protocol was previously known as the portmapper protocol.
 - s** Display a concise list of all registered RPC programs on *host*. If *host* is not specified, it defaults to the local host.
 - t** Make an RPC call to procedure 0 of *prognum* on the specified *host* using TCP, and report whether a response was received. This option is made obsolete by the **-T** option as shown in the third synopsis.
 - u** Make an RPC call to procedure 0 of *prognum* on the specified *host* using UDP, and report whether a response was received. This option is made obsolete by the **-T** option as shown in the third synopsis.

EXAMPLES

To show all of the RPC services registered on the local machine use:

```
example% rpcinfo
```

To show all of the RPC services registered with **rpcbind** on the machine named **klaxon** use:

```
example% rpcinfo klaxon
```

The information displayed by the above commands can be quite lengthy. Use the **-s** option to display a more concise list:

```
example% rpcinfo -s klaxon
```

program	version(s)	netid(s)	service	owner
100000	2,3,4	tcp,udp,ticlts,ticots,ticotsord	rpcbind	super-user
100008	1	ticotsord,ticots,ticlts,udp,tcp	walld	super-user
100002	2,1	ticotsord,ticots,ticlts,udp,tcp	rusersd	super-user
100001	2,3,4	ticotsord,ticots,tcp,ticlts,udp	rstatd	super-user
100012	1	ticotsord,ticots,ticlts,udp,tcp	sprayd	super-user
100007	3	ticotsord,ticots,ticlts,udp,tcp	ypbind	super-user
100029	1	ticotsord,ticots,ticlts	keyserv	super-user
100078	4	ticotsord,ticots,ticlts	kerbd	super-user
100024	1	ticotsord,ticots,ticlts,udp,tcp	status	super-user
100021	2,1	ticotsord,ticots,ticlts,udp,tcp	nlockmgr	super-user
100020	1	ticotsord,ticots,ticlts,udp,tcp	llockmgr	super-user

To show whether the RPC service with program number *prognum* and version *versnum* is registered on the machine named **klaxon** for the transport TCP use:

```
example% rpcinfo -T tcp klaxon prognum versnum
```

To show all RPC services registered with version 2 of the **rpcbind** protocol on the local machine use:

```
example% rpcinfo -p
```

To delete the registration for version 1 of the **walld** (program number **100008**) service for all transports use:

```
example# rpcinfo -d 100008 1
```

or

```
example# rpcinfo -d walld 1
```

SEE ALSO

rpcbind(1M), **rpc(3N)**, **netconfig(4)**, **rpc(4)**

NAME	rpld – x86 Network Booting RPL (Remote Program Load) Server
SYNOPSIS	<code>/usr/sbin/rpld [-fdMblgz] interface</code> <code>/usr/sbin/rpld -a [-fdMblgz]</code>
AVAILABILITY	x86 SUNWcsu
DESCRIPTION	<p>The RPL server provides network booting functionality to x86 clients by listening to boot requests from them according to the RPL protocol specifications. Boot requests can be generated by clients using the boot floppy supplied in the x86 distribution. Once the request has been received, the server validates the client and adds it to its internal service list. Subsequent requests from the client to download bootfiles will result in the sending of data frames from the server to the client specifying where to load the boot program in memory. When all the bootfiles have been downloaded, the server specifies where to start execution to initiate the boot process.</p> <p>In the first synopsis, the interface parameter names the network interface upon which rpld is to listen for requests. For example:</p> <pre style="margin-left: 40px;">/usr/sbin/rpld /dev/le0 /usr/sbin/rpld /dev/smc0</pre> <p>In the second synopsis, rpld locates all of the network interfaces present on the system and starts a daemon process for each one.</p> <p>The server starts by reading the default configuration file, or an alternate configuration file if one is specified. If no configuration file can be found, internal default values will be used. Alternatively, command line options are available to override any of the values in the configuration file. After the configuration options are set, it then opens the network interface as specified in the command line and starts listening to RPL boot requests.</p> <p>Network boot x86 clients have to have information pre-configured on a server for the RPL server to validate and serve them. This involves putting configuration information in both the ethers(4) and the bootparams(4) databases. The ethers database contains a translation from the physical node address to the IP address of the clients and is normally used by the RARP server. The bootparams database stores all other information needed for booting off this client, such as the number of bootfiles and the file names of the various boot components. Both databases can be looked up by the RPL server through NIS. See the sub-section Client Configuration for information on how to set up these databases.</p> <p>To assist in the administration and maintenance of the network boot activity, there are two run-time signals that the server will accept to change some run-time parameters and print out useful status information. See the sub-section Signals for details.</p> <p>The RPL server is not limited to the ability to boot only x86 clients. If properly configured, the server should be able to download any bootfiles to the clients.</p>

Client Configuration

The following configuration information is specific to booting x86 clients.

In order to allow clients to boot x86 from across the network, the client's information has to be pre-configured in two databases: **ethers(4)** and **bootparams(4)**. Both databases can be accessed through NIS. Refer to *x86: Installing Solaris Software* for information on how to configure a diskless x86 client. The discussion contained in the rest of this section is provided for your information only and should not be performed manually.

The **ethers** database contains a translation table to convert the physical node address to the IP address of the client. Therefore, an IP address must be assigned to the client (if this has not been done already), the node address of the client must be obtained, and then this information needs to be entered in the **ethers** database.

The bulk of the configuration is done in the **bootparams** database. This is a free-format database that essentially contains a number of keyword-value string pairs. A number of keywords have been defined for specific purposes, like the **bootparams** RPC in **bootparamd(1M)**. Three more keywords have been defined for the RPL server. They are **numbootfiles**, **bootfile**, and **bootaddr**. All three keywords must be in lowercase letters with no spaces before or after the equals symbol following the keyword.

numbootfiles Specifies the number of files to be downloaded to the network boot client. The format of this option is:

numbootfiles=n

Always use **numbootfiles=3** to boot x86 across the network.

bootfile Specifies the path name of the bootfile to be downloaded and where in memory to start loading the bootfile. A complete path name should be used. For example, assuming the client's IP address is 129.181.32.15:

bootfile=/rplboot/129.181.32.15.hw.com:45000

bootfile=/rplboot/129.181.32.15.glue.com:35000

bootfile=/rplboot/129.181.32.15.inetboot=8000

The path name following the equals symbol specifies the bootfile to be downloaded, and the hex address following the colon (:) is the absolute address of the memory location to start loading that bootfile. These addresses should be in the range of 7c00 to a0000 (i.e., the base 640K range excluding the interrupt vector and BIOS data areas). Address 45000 for this **hw.com** bootfile is also a suggested value and if possible should not be changed. The address of 35000 for **glue.com** is a suggested value that, if possible, should not be changed. The address of 8000 for **inetboot** is an absolute requirement and should never be changed.

These files, when created following the procedures in the *x86: Installing Solaris Software*, are actually symbolic links to the real file to be downloaded to the client. **hw.com** is linked to a special driver that corresponds to the network interface card of the client. **glue.com** and **inetboot** are generic to all network boot clients.

The order of these bootfile lines is not significant, but because problems have been found with certain boot PROMs, it is highly recommended that the bootfile lines be ordered in descending order of the load addresses.

bootaddr

The absolute address in memory to start executing after all the bootfiles have been downloaded. This address should always correspond to the address where **glue.com** is being loaded. If possible, always use:

bootaddr=35000

OPTIONS

-f config filename.

Use this to specify a configuration file name other than the system default **/etc/rpld.conf** file.

-d debug level.

Specify a level of **0** if you do not want any error or warning messages to be generated, or a level from **1** to **9** for increasing amounts of messages. This option corresponds to the *DebugLevel* setting in the configuration file. The default value is **0**. Note that it is best to limit the level to **8** or below; use of level **9** may generate so many debug messages that the performance of the RPL server may be impacted.

-D debug destination.

Specify **0** to send error or warning messages to standard output, **1** to syslogd, and **2** to the log file. This option corresponds to the *DebugDest* setting in the configuration file. The default value is **2**.

-M maximum clients.

Specify the maximum number of simultaneous network boot clients to be served. This option corresponds to the *MaxClients* setting in the configuration file. A value of **-1** means unlimited, and the actual number will depend on available system resources. The default value is **-1**.

-b background mode.

Specify **1** to run the server in the background and relinquish the controlling terminal, or **0** to run in the foreground without relinquishing the controlling terminal. This option corresponds to the *BackGround* setting in the configuration file. If you have specified that the error or warning messages be sent to standard output in the configuration file or by using the **-D** option above, the server cannot be run in background mode. Doing so will cause the server to exit after announcing the error.

-l log filename.

Specify an alternate log file name to hold the error or warning messages in connection with the **-D 2** option or the configuration file *DebugDest = 2* setting. This option corresponds to the *LogFile* setting in the configuration file. The default is **/var/spool/rpld.log**.

-s start delay count.

This option corresponds to the *StartDelay* setting in the configuration file. Specify the number of delay units between outgoing data frames sent to clients to avoid retransmission requests from them. Using the LLC type 1 protocol, data transfer is a one-way, best-effort delivery mechanism. The server, without any type of delay mechanism, can overrun the client by sending data frames too quickly. Therefore, a variable delay is built into the server to limit the speed of sending data to the clients, thus avoiding the clients sending back retransmission requests. This value should be machine environment specific. If you have a fast server machine but slow client machines, you may want to set a large start delay count. If you have comparable server and client machines, the delay count may be set to **1**. The delay is only approximate and should not be taken as an accurate measure of time. There is no specific correlation between the delay unit and the actual time of delay. The default value is **20**.

-g delay granularity.

This corresponds to the *DelayGran* setting in the configuration file. If retransmission requests from clients do occur, the delay granularity factor will be used to adjust the delay count for this client upwards or downwards. If the retransmission request is caused by data overrun, the delay count will be incremented by delay granularity units to increase the delay between data frames. If the retransmission request is caused by sending data too slowly, this will be used to adjust the delay count downwards to shorten the delay. Eventually the server will settle at the delay count value that works best with the speed of the client and no retransmission request will be needed. The default value is **2**.

-z frame size.

This option corresponds to the *FrameSize* setting in the configuration file. This specifies the size of the data frames used to send data to the clients. This is limited by the underlying physical medium. For **ethernet/802.3**, the maximum physical frame size is 1500 octets. The default value is **1500**. Note that the protocol overhead of LLC1 and RPL is 32 octets, resulting in a maximum data length of 1468 octets.

Signals

The RPL server accepts two signals to change run-time parameters and display status information, respectively:

HANGUP

This will cause the RPL server to reread the default configuration file **/etc/rpld.conf** or an alternate configuration file if one is specified when the server is started. New values of certain parameters can be used immediately, such as *DebugLevel*, *DebugDest*, *LogFile*, *DelayGran*, and *FrameSize*. For *MaxClients*, if the server is already serving more than the new value, the server will not accept additional boot requests until the number has fallen below the *MaxClients* parameter. For *StartDelay*, this will only affect new boot requests. All the existing delay counts for the various clients in service will not be affected. Finally, the *BackGround* parameter will have no effect once the server has been running. You

cannot change the mode of service without first killing the server and then restarting it.

USR1 This signal will cause the server to dump all the parameter values and the status of each individual boot client to the destination specified by *DebugDest*.

FILES **/usr/sbin/rpld**
/etc/rpld.conf
/var/spool/rpld.log
/etc/ethers
/etc/bootparams
/rplboot

SEE ALSO **bootparamd(1M), in.rarpd(1M), bootparams(4), ethers(4), nsswitch.conf(4), rpld.conf(4)**
x86: Installing Solaris Software

NAME	rquotad – remote quota server
SYNOPSIS	/usr/lib/nfs/rquotad
AVAILABILITY	SUNWcsu
DESCRIPTION	rquotad is an rpc(4) server which returns quotas for a user of a local file system which is mounted by a remote machine over the NFS. The results are used by quota(1M) to display user quotas for remote file systems. The rquotad daemon is normally invoked by inetd(1M) .
FILES	quotas quota file at the file system root
SEE ALSO	inetd(1M) , quota(1M) , rpc(4) , services(4) <i>SPARC: Installing Solaris Software</i> <i>x86: Installing Solaris Software</i>

NAME	rsh, restricted_shell – restricted shell command interpreter
SYNOPSIS	<code>/usr/lib/rsh [-acefhiknprstuvx] [argument...]</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>rsh is a limiting version of the standard command interpreter sh, used to restrict logins to execution environments whose capabilities are more controlled than those of sh (see sh(1) for complete description and usage).</p> <p>When the shell is invoked, it scans the environment for the value of the environmental variable, SHELL. If it is found and rsh is the file name part of its value, the shell becomes a restricted shell.</p> <p>The actions of rsh are identical to those of sh, except that the following are disallowed:</p> <ul style="list-style-type: none"> changing directory (see cd(1)), setting the value of \$PATH, specifying path or command names containing /, redirecting output (> and >>). <p>The restrictions above are enforced after <i>.profile</i> is interpreted.</p> <p>A restricted shell can be invoked in one of the following ways:</p> <ol style="list-style-type: none"> (1) rsh is the file name part of the last entry in the /etc/passwd file (see passwd(4)); (2) the environment variable SHELL exists and rsh is the file name part of its value; the environment variable SHELL needs to be set in the .login file; (3) the shell is invoked and rsh is the file name part of argument 0; (4) the shell is invoked with the -r option. <p>When a command to be executed is found to be a shell procedure, rsh invokes sh to execute it. Thus, it is possible to provide to the end-user shell procedures that have access to the full power of the standard shell, while imposing a limited menu of commands; this scheme assumes that the end-user does not have write and execute permissions in the same directory.</p> <p>The net effect of these rules is that the writer of the <i>.profile</i> (see profile(4)) has complete control over user actions by performing guaranteed setup actions and leaving the user in an appropriate directory (probably <i>not</i> the login directory).</p> <p>The system administrator often sets up a directory of commands (that is, /usr/rbin) that can be safely invoked by a restricted shell. Some systems also provide a restricted editor, red.</p>
EXIT CODES	Errors detected by the shell, such as syntax errors, cause the shell to return a non-zero exit status. If the shell is being used non-interactively execution of the shell file is abandoned. Otherwise, the shell returns the exit status of the last command executed.

SEE ALSO

cd(1), intro(1), login(1), sh(1), exec(2), passwd(4), profile(4)

NOTES

The restricted shell, **/usr/lib/rsh**, should not be confused with the remote shell, **/usr/bin/rsh**, which is documented in **rsh(1)**.

NAME	rtc – provide all real-time clock and GMT-lag management	
SYNOPSIS	rtc [-c] [-z <i>zone-name</i>]	
AVAILABILITY	x86 SUNWcsu	
DESCRIPTION	<p>The /usr/sbin/rtc command is used to reconcile the difference in the way time is established between UNIX systems, which utilize Greenwich Mean Time (GMT), and MS-DOS systems, which utilize local time. When used with no arguments, rtc displays the currently configured time zone string which was recorded when rtc -z <i>zone-name</i> was last run.</p> <p>The rtc command is not normally run from a shell prompt; it is generally invoked by the system. Commands such as date(1) and rdate(1M), which are used to set the time on a system, invoke /usr/sbin/rtc -c to ensure that daylight savings time (DST) is corrected for properly.</p>	
OPTIONS	-c	<p>This option checks for DST and makes corrections if necessary. It is normally run once a day by a <i>cron</i> job.</p> <p>If there is no RTC time zone or /etc/rtc_config file, this option will do nothing.</p>
	-z <i>zone-name</i>	<p>This option, which is normally run by the system at software installation time, is used to specify the time zone in which the RTC is to be maintained. It updates the configuration file /etc/rtc_config with the name of the specified zone and the current GMT lag for that zone. If there is an existing rtc_config file, this command will update it. If not, this command will create it.</p>
FILES	/usr/sbin/rtc	The command used to provide all real-time clock and GMT lag management.
	/etc/rtc_config	The data file used to record the time zone and GMT lag. This file is completely managed by /usr/sbin/rtc , and it is read by the kernel.
SEE ALSO	date(1) , rdate(1M)	

NAME	runacct – run daily accounting
SYNOPSIS	<code>/usr/lib/acct/runacct [mmdd [state]]</code>
DESCRIPTION	<p>runacct is the main daily accounting shell procedure. It is normally initiated using cron. runacct processes connect, fee, disk, and process accounting files. It also prepares summary files for prdaily or billing purposes. runacct is distributed only to source code licensees.</p> <p>runacct takes care not to damage active accounting files or summary files in the event of errors. It records its progress by writing descriptive diagnostic messages into active. When an error is detected, a message is written to /dev/console, mail (see mail(1)) is sent to root and adm, and runacct terminates. runacct uses a series of lock files to protect against re-invocation. The files lock and lock1 are used to prevent simultaneous invocation, and lastdate is used to prevent more than one invocation per day.</p> <p>runacct breaks its processing into separate, restartable <i>states</i> using statefile to remember the last <i>state</i> completed. It accomplishes this by writing the <i>state</i> name into statefile. runacct then looks in statefile to see what it has done and to determine what to process next. <i>states</i> are executed in the following order:</p> <p>SETUP Move active accounting files into working files.</p> <p>WTMPFIX Verify integrity of wtmp file, correcting date changes if necessary.</p> <p>CONNECT Produce connect session records in tacct.h format.</p> <p>PROCESS Convert process accounting records into tacct.h format.</p> <p>MERGE Merge the connect and process accounting records.</p> <p>FEES Convert output of chargefee into tacct.h format, merge with connect, and process accounting records.</p> <p>DISK Merge disk accounting records with connect, process, and fee accounting records.</p> <p>MERGETACCT Merge the daily total accounting records in daytacct with the summary total accounting records in /var/adm/acct/sum/tacct.</p> <p>CMS Produce command summaries.</p> <p>USEREXIT Any installation dependent accounting programs can be included here.</p> <p>CLEANUP Clean up temporary files and exit.</p> <p>To restart runacct after a failure, first check the active file for diagnostics, then fix any corrupted data files, such as pacct or wtmp. The lock, lock1, and lastdate files must be removed before runacct can be restarted. The argument <i>mmdd</i> is necessary if runacct is being restarted. <i>mmdd</i> specifies the month and day for which runacct will rerun the accounting. The entry point for processing is based on the contents of statefile; to override this, include the desired <i>state</i> on the command line to designate where processing should begin.</p>

EXAMPLES

To start **runacct**:

example% nohup runacct 2> /var/adm/acct/nite/fd2log &

To restart **runacct**:

example% nohup runacct 0601 2>> /var/adm/acct/nite/fd2log &

To restart **runacct** at a specific *state*:

example% nohup runacct 0601 MERGE 2>> /var/adm/acct/nite/fd2log &

FILES

/var/adm/wtmp
/var/adm/pacctincr
/var/adm/acct/nite/active
/var/adm/acct/nite/daytacct
/var/adm/acct/nite/lock
/var/adm/acct/nite/lock1
/var/adm/acct/nite/lastdate
/var/adm/acct/nite/statefile

SEE ALSO

acctcom(1), mail(1), acct(1M), acctcms(1M), acctcon(1M), acctmerg(1M), acctprc(1M), acctsh(1M), cron(1M), fwtmp(1M), acct(2), acct(4), utmp(4)

NOTES

It is not recommended to restart **runacct** in the **SETUP** *state*. Run **SETUP** manually and restart using:

runacct mddd WTMPFIX

If **runacct** failed in the **PROCESS** *state*, remove the last **ptacct** file because it will not be complete.

The **runacct** command can process a maximum of

6000 distinct sessions
1000 distinct terminal lines
2000 distinct login names

during a single invocation of the command. If at some point the actual number of any one of these items exceeds the maximum, the command will not succeed.

NAME	rwall – write to all users over a network
SYNOPSIS	<i>/usr/sbin/rwall hostname ...</i> <i>/usr/sbin/rwall -n netgroup ...</i> <i>/usr/sbin/rwall -h hostname -n netgroup</i>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>rwall reads a message from standard input until EOF. It then sends this message, preceded by the line:</p> <p style="text-align: center;">Broadcast Message ...</p> <p>to all users logged in on the specified host machines. With the -n option, it sends to the specified network groups.</p>
OPTIONS	<p>-n netgroup Send the broadcast message to the specified network groups.</p> <p>-h hostname Specify the <i>hostname</i>, the name of the host machine.</p>
SEE ALSO	inetd(1M) , listen(1M) , pmadm(1M) , sacadm(1M) , wall(1M)
NOTES	The timeout is fairly short to allow transmission to a large group of machines (some of which may be down) in a reasonable amount of time. Thus the message may not get through to a heavily loaded machine.

NAME	sac – service access controller								
SYNOPSIS	sac <i>-t sanity_interval</i> <i>/usr/lib/saf/sac</i>								
DESCRIPTION	<p>The Service Access Controller (SAC) is the overseer of the server machine. It is started when the server machine enters multiuser mode. The SAC performs several important functions as explained below.</p> <p><i>Customizing the SAC environment.</i> When sac is invoked, it first looks for the per-system configuration script <i>/etc/saf/_sysconfig</i>. sac interprets <i>_sysconfig</i> to customize its own environment. The modifications made to the SAC environment by <i>_sysconfig</i> are inherited by all the children of the SAC. This inherited environment may be modified by the children.</p> <p><i>Starting port monitors.</i> After it has interpreted the <i>_sysconfig</i> file, the sac reads its administrative file <i>/etc/saf/_sactab</i>. <i>_sactab</i> specifies which port monitors are to be started. For each port monitor to be started, sac forks a child (see fork(2)) and creates a utmp entry with the <i>type</i> field set to LOGIN_PROCESS. Each child then interprets its per-port monitor configuration script <i>/etc/saf/pmtag/_config</i>, if the file exists. These modifications to the environment affect the port monitor and will be inherited by all its children. Finally, the child process execs the port monitor, using the command found in the <i>_sactab</i> entry. (See sacadm; this is the command given with the <i>-c</i> option when the port monitor is added to the system.)</p> <p><i>Polling port monitors to detect failure.</i> The <i>-t</i> option sets the frequency with which sac polls the port monitors on the system. This time may also be thought of as half of the maximum latency required to detect that a port monitor has failed and that recovery action is necessary.</p> <p><i>Administrative functions.</i> The Service Access Controller represents the administrative point of control for port monitors. Its administrative tasks are explained below.</p> <p>When queried (sacadm with either <i>-I</i> or <i>-L</i>), the Service Access Controller returns the status of the port monitors specified, which sacadm prints on the standard output. A port monitor may be in one of six states:</p> <table border="0" style="margin-left: 40px;"> <tr> <td style="padding-right: 20px;">ENABLED</td> <td>The port monitor is currently running and is accepting connections. See sacadm(1M) with the <i>-e</i> option.</td> </tr> <tr> <td>DISABLED</td> <td>The port monitor is currently running and is not accepting connections. See sacadm with the <i>-d</i> option, and see NOTRUNNING, below.</td> </tr> <tr> <td>STARTING</td> <td>The port monitor is in the process of starting up. STARTING is an intermediate state on the way to ENABLED or DISABLED.</td> </tr> <tr> <td>FAILED</td> <td>The port monitor was unable to start and remain running.</td> </tr> </table>	ENABLED	The port monitor is currently running and is accepting connections. See sacadm(1M) with the <i>-e</i> option.	DISABLED	The port monitor is currently running and is not accepting connections. See sacadm with the <i>-d</i> option, and see NOTRUNNING , below.	STARTING	The port monitor is in the process of starting up. STARTING is an intermediate state on the way to ENABLED or DISABLED .	FAILED	The port monitor was unable to start and remain running.
ENABLED	The port monitor is currently running and is accepting connections. See sacadm(1M) with the <i>-e</i> option.								
DISABLED	The port monitor is currently running and is not accepting connections. See sacadm with the <i>-d</i> option, and see NOTRUNNING , below.								
STARTING	The port monitor is in the process of starting up. STARTING is an intermediate state on the way to ENABLED or DISABLED .								
FAILED	The port monitor was unable to start and remain running.								

- STOPPING** The port monitor has been manually terminated but has not completed its shutdown procedure. **STOPPING** is an intermediate state on the way to **NOTRUNNING**.
- NOTRUNNING** The port monitor is not currently running. (See **sacadm** with **-k**.) This is the normal “not running” state. When a port monitor is killed, all ports it was monitoring are inaccessible. It is not possible for an external user to tell whether a port is not being monitored or the system is down. If the port monitor is not killed but is in the **DISABLED** state, it may be possible (depending on the port monitor being used) to write a message on the inaccessible port telling the user who is trying to access the port that it is disabled. This is the advantage of having a **DISABLED** state as well as the **NOTRUNNING** state.

When a port monitor terminates, the SAC removes the **utmp** entry for that port monitor. The SAC receives all requests to enable, disable, start, or stop port monitors and takes the appropriate action.

The SAC is responsible for restarting port monitors that terminate. Whether or not the SAC will restart a given port monitor depends on two things:

- The restart count specified for the port monitor when the port monitor was added by **sacadm**; this information is included in **/etc/saf/pmtag/_sactab**.
- The number of times the port monitor has already been restarted.

- OPTIONS** **-t** *sanity_interval* Sets the frequency (*sanity_interval*) with which **sac** polls the port monitors on the system.
- FILES** **/etc/saf/_sactab**
/etc/saf/_sysconfig
/var/adm/utmp
/var/saf/_log
- SEE ALSO** **pmadm(1M)**, **sacadm(1M)**, **fork(2)**

NAME	sacadm – service access controller administration
SYNOPSIS	<pre> sacadm -a -p pmtag -t type -c cmd -v ver [-f dx] [-n count] [-y comment] [-z script] sacadm -r -p pmtag sacadm -s -p pmtag sacadm -k -p pmtag sacadm -e -p pmtag sacadm -d -p pmtag sacadm -l [-p pmtag -t type] sacadm -L [-p pmtag -t type] sacadm -g -p pmtag [-z script] sacadm -G [-z script] sacadm -x [-p pmtag] </pre>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>sacadm is the administrative command for the upper level of the Service Access Facility hierarchy (port monitor administration). sacadm performs the following functions:</p> <ul style="list-style-type: none"> • adds or removes a port monitor • starts or stops a port monitor • enables or disables a port monitor • installs or replaces a per-system configuration script • installs or replaces a per-port monitor configuration script • prints requested port monitor information <p>Requests about the status of port monitors (-l and -L) and requests to print per-port monitor and per-system configuration scripts (-g and -G without the -z option) may be executed by any user on the system. Other sacadm commands may be executed only by the super-user.</p>
OPTIONS	<p>-a Add a port monitor. When adding a port monitor, sacadm creates the supporting directory structure in /etc/saf and /var/saf and adds an entry for the new port monitor to /etc/saf/_sactab. The file _sactab already exists on the delivered system. Initially, it is empty except for a single line, which contains the version number of the Service Access Controller.</p> <p>Unless the command line that adds the new port monitor includes the -f option with the -x argument, the new port monitor will be started. Because of the complexity of the options and arguments that follow the -a option, it may be convenient to use a command script or the menu system to add port monitors.</p> <p>-c cmd Execute the command string <i>cmd</i> to start a port monitor. The -c option may be used only with a -a. A -a option requires a -c.</p>

- d** Disable the port monitor *pmtag*.
- e** Enable the port monitor *pmtag*.
- f dx** The **-f** option specifies one or both of the following two flags which are then included in the flags field of the **_sactab** entry for the new port monitor. If the **-f** option is not included on the command line, no flags are set and the default conditions prevail. By default, a port monitor is started. A **-f** option with no following argument is illegal.
- d** Do not enable the new port monitor.
 - x** Do not start the new port monitor.
- g** The **-g** option is used to request output or to install or replace the per-port monitor configuration script */etc/saf/pmtag/_config*. **-g** requires a **-p** option. The **-g** option with only a **-p** option prints the per-port monitor configuration script for port monitor *pmtag*. The **-g** option with a **-p** option and a **-z** option installs the file **script** as the per-port monitor configuration script for port monitor *pmtag*. Other combinations of options with **-g** are invalid.
- G** The **-G** option is used to request output or to install or replace the per-system configuration script */etc/saf/_sysconfig*. The **-G** option by itself prints the per-system configuration script. The **-G** option in combination with a **-z** option installs the file **script** as the per-system configuration script. Other combinations of options with a **-G** option are invalid.
- k** Stop port monitor *pmtag*.
- l** The **-l** option is used to request port monitor information. The **-l** by itself lists all port monitors on the system. The **-l** option in combination with the **-p** option lists only the port monitor specified by *pmtag*. A **-l** in combination with the **-t** option lists all port monitors of type *type*. Any other combination of options with the **-l** option is invalid.
- L** The **-L** option is identical to the **-l** option except that the output appears in a condensed format.
- n count** Set the restart count to *count*. If a restart count is not specified, count is set to **0**. A count of **0** indicates that the port monitor is not to be restarted if it fails.
- p pmtag** Specifies the tag associated with a port monitor.

- r** Remove port monitor *pmtag*. **sacadm** removes the port monitor entry from `/etc/saf/_sactab`. If the removed port monitor is not running, then no further action is taken. If the removed port monitor is running, the Service Access Controller (SAC) sends it **SIGTERM** to indicate that it should shut down. Note that the port monitor's directory structure remains intact.
- s** Start a port monitor. The SAC starts the port monitor *pmtag*.
- t type** Specifies the port monitor type.
- v ver** Specifies the version number of the port monitor. This version number may be given as
 - `-v `pmspec -V``
 where *pmspec* is the special administrative command for port monitor *pmtag*. This special command is **ttyadm** for **ttymon** and **nlsadmin** for **listen**. The version stamp of the port monitor is known by the command and is returned when *pmspec* is invoked with a `-V` option.
- x** The `-x` option by itself tells the SAC to read its database file (`_sactab`). The `-x` option with the `-p` option tells port monitor *pmtag* to read its administrative file.
- y comment** Include *comment* in the `_sactab` entry for port monitor *pmtag*.
- z script** Used with the `-g` and `-G` options to specify the name of a file that contains a configuration script. With the `-g` option, *script* is a per-port monitor configuration script; with `-G` it is a per-system configuration script. Modifying a configuration script is a three-step procedure. First a copy of the existing script is made (`-g` or `-G`). Then the copy is edited. Finally, the copy is put in place over the existing script (`-g` or `-G` with `-z`).

OUTPUT

If successful, **sacadm** will exit with a status of **0**. If **sacadm** fails for any reason, it will exit with a nonzero status. Options that request information will write the information on the standard output. In the condensed format (`-L`), port monitor information is printed as a sequence of colon-separated fields; empty fields are indicated by two successive colons. The standard format (`-I`) prints a header identifying the columns, and port monitor information is aligned under the appropriate headings. In this format, an empty field is indicated by a hyphen. The comment character is `#`.

EXAMPLES

The following command line adds a port monitor. The port monitor tag is **npack**; its type is **listen**; if necessary, it will restart three times before failing; its administrative command is **nlsadmin**; and the configuration script to be read is in the file **script**:

```
sacadm -a -p npack -t listen -c /usr/lib/saf/listen npack  
-v `nlsadmin -V` -n 3 -z script
```

Remove a port monitor whose tag is **pmtag**:

sacadm -r -p pmtag

Start the port monitor whose tag is **pmtag**:

sacadm -s -p pmtag

Stop the port monitor whose tag is **pmtag**:

sacadm -k -p pmtag

Enable the port monitor whose tag is **pmtag**:

sacadm -e -p pmtag

Disable the port monitor whose tag is **pmtag**:

sacadm -d -p pmtag

List status information for all port monitors:

sacadm -l

List status information for the port monitor whose tag is **pmtag**:

sacadm -l -p pmtag

List the same information in condensed format:

sacadm -L -p pmtag

List status information for all port monitors whose type is **listen**:

sacadm -l -t listen

Replace the per-port monitor configuration script associated with the port monitor whose tag is **pmtag** with the contents of the file **file.config**:

sacadm -g -p pmtag -z file.config

FILES

/etc/saf/_sactab

/etc/saf/_sysconfig

/etc/saf/pmtag/_config

SEE ALSO

pmadm(1M), **sac(1M)**, **doconfig(3N)**

NAME	sadmind – distributed system administration daemon								
SYNOPSIS	sadmind [<i>-c keywords</i>] [<i>-i secs</i>] [<i>-l [logfile]</i>] [<i>-O OW_path_name</i>] [<i>-S security_level</i>] [<i>-v</i>]								
AVAILABILITY	SUNWadmfw								
DESCRIPTION	<p>sadmind is the daemon used by Solstice AdminSuite applications to perform distributed system administration operations.</p> <p>The sadmind daemon is started automatically by the inetd daemon whenever a request to invoke an operation is received. The sadmind daemon process continues to run for 15 minutes after the last request is completed, unless a different idle-time is specified with the <i>-i</i> command line option. The sadmind daemon may be started independently from the command line, for example, at system boot time. In this case, the <i>-i</i> option has no effect; sadmind continues to run, even if there are no active requests.</p> <p>The sadmind daemon process can be configured to write tracing information into a log file by specifying the <i>-c</i> and <i>-l</i> command line options. The <i>-c</i> option specifies a comma-separated list of keywords indicating the types of information to be logged. The following keywords may be useful to you as an administrator:</p> <table border="0"> <tr> <td style="padding-right: 20px;">System-Info</td> <td>Includes messages about when the sadmind daemon was started and stopped.</td> </tr> <tr> <td>Requests</td> <td>Includes messages about which operations sadmind invoked and when.</td> </tr> <tr> <td>Errors</td> <td>Includes messages about errors that occurred during the daemon execution.</td> </tr> <tr> <td>*</td> <td>Includes all possible log messages.</td> </tr> </table> <p>The <i>-l</i> option enables logging and optionally specifies the path and file name of the log file. If no log file is specified, the default log file <code>/var/adm/admin.log</code> is used.</p>	System-Info	Includes messages about when the sadmind daemon was started and stopped.	Requests	Includes messages about which operations sadmind invoked and when.	Errors	Includes messages about errors that occurred during the daemon execution.	*	Includes all possible log messages.
System-Info	Includes messages about when the sadmind daemon was started and stopped.								
Requests	Includes messages about which operations sadmind invoked and when.								
Errors	Includes messages about errors that occurred during the daemon execution.								
*	Includes all possible log messages.								
OPTIONS	<p>The following command line options may be useful to you as an administrator when starting the sadmind daemon:</p> <table border="0"> <tr> <td style="padding-right: 20px;"><i>-c keywords</i></td> <td>Specify the types of information to be logged as a comma-separated list of keywords. The default is to log all types of messages.</td> </tr> <tr> <td><i>-i secs</i></td> <td>Specify the number of seconds for sadmind to stay up after the last request is completed. The default is 15 minutes (900 seconds). If <i>secs</i> is 0 or over 10,000,000, sadmind stays up forever. <i>-i</i> only applies when sadmind is started by the inetd daemon. You may want sadmind to run permanently (or for extended durations) on systems that are frequently administered by applications using sadmind (for example, a server managed through Host Manager) to improve application performance.</td> </tr> </table>	<i>-c keywords</i>	Specify the types of information to be logged as a comma-separated list of keywords. The default is to log all types of messages.	<i>-i secs</i>	Specify the number of seconds for sadmind to stay up after the last request is completed. The default is 15 minutes (900 seconds). If <i>secs</i> is 0 or over 10,000,000, sadmind stays up forever. <i>-i</i> only applies when sadmind is started by the inetd daemon. You may want sadmind to run permanently (or for extended durations) on systems that are frequently administered by applications using sadmind (for example, a server managed through Host Manager) to improve application performance.				
<i>-c keywords</i>	Specify the types of information to be logged as a comma-separated list of keywords. The default is to log all types of messages.								
<i>-i secs</i>	Specify the number of seconds for sadmind to stay up after the last request is completed. The default is 15 minutes (900 seconds). If <i>secs</i> is 0 or over 10,000,000, sadmind stays up forever. <i>-i</i> only applies when sadmind is started by the inetd daemon. You may want sadmind to run permanently (or for extended durations) on systems that are frequently administered by applications using sadmind (for example, a server managed through Host Manager) to improve application performance.								

- l**[logfile] Enable logging and optionally define the path name to the distributed system administration log file. The default log file is:
/var/adm/admin.log
- O** *OW_path_name* Define the path name to the OpenWindows home directory. If this option is not specified, the **sadmind** daemon will use the OpenWindows home directory defined in the **OPENWINHOME** environment variable, if defined; the home directory specified in the **/etc/OPENWINHOME** file, if it exists; or the default directory **/usr/openwin**. When the **sadmind** daemon is started by the **inetd** daemon, the environment variable **OPENWINHOME** is typically not defined. If the OpenWindows home directory is not one of the path names specified (**/usr/openwin** or in the file **/etc/OPENWINHOME**), the **-O** option must be added to the **sadmind** entry in the **inetd.conf(4)** configuration file.
- S** *security_level* Define the level of security to be used by the **sadmind** daemon when checking a client's right to perform an operation on the server system. Security level specifies the authentication mechanism used to provide and check the client's identity. The client's identity must be authenticated by the specified mechanism for **sadmind** to accept his or her request. The system-wide authentication requirements set by the security level may take precedence over any operation-specific requirements. Consequently, the security level can be used system-wide to ensure that all operations meet minimum authentication requirements, regardless of the requirements assigned specifically to an operation. In addition, the security level determines whether **sadmind** will perform authorization access control checking.
- Security level may be one of the following:
- 0** Set authentication type to **NONE**. All clients' user and group identities are set to the **nobody** identity by **sadmind** (see *Solstice AdminSuite 2.1 User's Guide*). If access is granted to **nobody**, **sadmind** executes the operation. Use this level only for testing.
 - 1** Set authentication type to **WEAK**. Clients' user and group identities are set by **sadmind** from their authentication credentials. Client identities are accepted by **sadmind** when they have satisfied either **AUTH_SYS** or **AUTH_DES** authentication mechanisms. The authenticated client identity is checked by **sadmind** for authorization to execute the operation. If an operation calls for a stronger security level, **sadmind** demotes the user identity to **nobody**, and then checks whether **nobody** is authorized to execute the operation. Since **AUTH_SYS** client credentials are easily forged, this level should be used only in relatively secure environments. No check is done that the user ID of the client represents the same user on the server system

as on the client system. It is assumed that user and group identities are set up consistently on the network. This security level is the default.

- 2 Set authentication type to **STRONG**. Clients' user and group identities are set by **sadmind** from their authentication credential mappings (effectively, user and group IDs from **netid.byname** for NIS, or **cred** table for NIS+). Client identities are accepted by **sadmind** only when they have satisfied the **AUTH_DES** authentication mechanism. The **sadmind** daemon checks whether the client identity is authorized to execute the operation. This level provides the most secure environment for executing distributed administration operations. It overrides any weaker level specific to an operation. A **DES** credential must exist for the host running the **sadmind** daemon and all administration client user identities.

- v Enable the writing of log messages to the system logger, **syslogd**. Messages logged include fatal errors encountered while attempting to start the **sadmind** daemon process and those specified by the **-c** trace message keywords.

EXAMPLES

By default, the line in **/etc/inetd.conf** that starts **sadmind** appears as follows:

```
100232/10 tli rpc/udp wait root /usr/sbin/sadmind sadmind
```

To make a network as secure as possible, change the line to:

```
100232/10 tli rpc/udp wait root /usr/sbin/sadmind sadmind -S 2
```

To minimize delays due to starting up **sadmind**, change the line to include the **-i** option:

```
100232/10 tli rpc/udp wait root /usr/sbin/sadmind sadmind -i 86400
```

In this example, the duration that **sadmind** remains up after the last operation request was completed is extended to 24 hours (86,400 seconds). Extending the timeout period may enhance performance on servers and workstations that frequently run or are administered by applications that use the **sadmind** daemon (for example, **Solstice AdminSuite** applications such as **Host Manager**).

FILES

/var/adm/admin.log distributed system administration default log file
/etc/inetd.conf internet servers database file

SEE ALSO

inetd(1M), **rpcbind(1M)**, **inetd.conf(4)**
Solstice AdminSuite 2.1 User's Guide

NOTES

Whenever **inetd** fails to start **sadmind**, re-register the RPC number for **sadmind**, 100232, with **rpcbind** by sending the **inetd** process a **SIGHUP** signal:

```
example% kill -HUP pid
```

or

```
example% kill -1
```

Sometimes **inetd** does not start **sadmind** in response to system administration requests, even though the **inetd.conf** file has the correct entry for the **sadmind** daemon. This can happen when **sadmind** is started manually from the command line and takes over the previous registration of the **sadmind** RPC number, 100232, by **inetd**. When the manually-started **sadmind** daemon is terminated, the **sadmind** RPC number, 100232, is de-registered with **rpcbind**. Consequently, system administration requests are ignored by **inetd**.

NAME	saf – Service Access Facility
AVAILABILITY	SUNWcsr
DESCRIPTION	<p>The SAF generalizes the procedures for service access so that login access on the local system and network access to local services are managed in similar ways. Under the SAF, systems may access services using a variety of port monitors, including ttymon, the listener, and port monitors written expressly for a user's application. The manner in which a port monitor observes and manages access ports is specific to the port monitor and not to any component of the SAF. Users may therefore extend their systems by developing and installing their own port monitors. One of the important features of the SAF is that it can be extended in this way by users.</p> <p>Relative to the SAF, a service is a process that is started. There are no restrictions on the functions a service may provide. The SAF consists of a controlling process, the service access controller (SAC), and two administrative levels corresponding to two levels in the supporting directory structure. The top administrative level is concerned with port monitor administration, the lower level with service administration. The SAC is documented in the sac(1M) man page. The administrative levels and associated utilities are documented in the <i>System Administration Guide - Volume II</i>. The requirements for writing port monitors and the functions a port monitor must perform to run under the SAF and the SAC are documented here.</p>
Port Monitors	<p>A port monitor is a process that is responsible for monitoring a set of homogeneous, incoming ports on a machine. A port monitor's major purpose is to detect incoming service requests and to dispatch them appropriately.</p> <p>A port is an externally seen access point on a system. A port may be an address on a network (TSAP or PSAP), a hardwired terminal line, an incoming phone line, etc. The definition of what constitutes a port is strictly a function of the port monitor itself.</p> <p>A port monitor performs certain basic functions. Some of these are required to conform to the SAF; others may be specified by the requirements and design of the port monitor itself. Port monitors have two main functions:</p> <ul style="list-style-type: none"> • Managing ports • Monitoring ports for indications of activity <p>Port Management</p> <p>The first function of a port monitor is to manage a port. The actual details of how a port is managed are defined by the person who defines the port monitor. A port monitor is not restricted to handling a single port; it may handle multiple ports simultaneously.</p> <p>Note: Some examples of port management are setting the line speed on incoming phone connections, binding an appropriate network address, reinitializing the port when the service terminates, outputting a prompt, etc.</p> <p>Activity Monitoring</p>

The second function of a port monitor is to monitor the port or ports for which it is responsible for indications of activity. Two types of activity may be detected.

- The first is an indication to the port monitor to take some port monitor-specific action. Pressing the break key to indicate that the line speed should be cycled is an example of a port monitor activity. Not all port monitors need to recognize and respond to the same indications. The indication used to attract the attention of the port monitor is defined by the person who defines the port monitor.
- The second is an incoming service request. When a service request is received, a port monitor must be able to determine which service is being requested from the port on which the request is received. Note that the same service may be available on more than one port.

Other Port Monitor Functions

This section briefly describes other port monitor functions.

Restricting Access to the System

A port monitor must be able to restrict access to the system without disturbing services that are still running. In order to do this, a port monitor must maintain two internal states: enabled and disabled. The port monitor starts in the state indicated by the ISTATE environment variable provided by the sac. See sac(1M) for details. Enabling or disabling a port monitor affects all ports for which the port monitor is responsible. If a port monitor is responsible for a single port, only that port will be affected. If a port monitor is responsible for multiple ports, the entire collection of ports will be affected. Enabling or disabling a port monitor is a dynamic operation: it causes the port monitor to change its internal state. The effect does not persist across new invocations of the port monitor. Enabling or disabling an individual port, however, is a static operation: it causes a change to an administrative file. The effect of this change will persist across new invocations of the port monitor.

Creating **utmp** Entries

Port monitors are responsible for creating **utmp** entries with the type field set to USER_PROCESS for services they start, if this action has been specified (that is, if **-fu** was specified in the **pmadm** command line that added the service). These **utmp** entries may in turn be modified by the service. When the service terminates, the **utmp** entry must be set to DEAD_PROCESS.

Port Monitor Process IDs and Lock Files

When a port monitor starts, it writes its process id into a file named **_pid** in the current directory and places an advisory lock on the file.

Changing the Service Environment: Running **doconfig**(3N)

Before invoking the service designated in the port monitor administrative file, **_pmtab**, a port monitor must arrange for the per-service configuration script to be run, if one exists, by calling the library function **doconfig**(3N). Because the per-service configuration script may specify the execution of restricted commands, as well as for other security reasons, port monitors are invoked with root

permissions. The details of how services are invoked are specified by the person who defines the port monitor.

Terminating a Port Monitor

A port monitor must terminate itself gracefully on receipt of the signal SIGTERM. The termination sequence is the following:

1. The port monitor enters the stopping state; no further service requests are accepted.
2. Any attempt to re-enable the port monitor will be ignored.
3. The port monitor yields control of all ports for which it is responsible. It must be possible for a new instantiation of the port monitor to start correctly while a previous instantiation is stopping.
4. The advisory lock on the process id file is released. Once this lock is released, the contents of the process id file are undefined and a new invocation of the port monitor may be started.

SAF Files

This section briefly covers the files used by the SAF.

The Port Monitor Administrative File

A port monitor's current directory contains an administrative file named **_pmtab**; **_pmtab** is maintained by the **pmadm** command in conjunction with a port monitor-specific administrative command.

Note: The port monitor administrative command for a listen port monitor is **nladmin**(1M); the port monitor administrative command for ttymon is **ttyadm**(1M). Any port monitor written by a user must be provided with an administrative command specific to that port monitor to perform similar functions.

Per-Service Configuration Files

A port monitor's current directory also contains the per-service configuration scripts, if they exist. The names of the per-service configuration scripts correspond to the service tags in the **_pmtab** file.

Private Port Monitor Files

A port monitor may create private files in the directory **/var/saf/tag**, where **tag** is the name of the port monitor. Examples of private files are log files or temporary files.

The SAC/Port Monitor Interface

The SAC creates two environment variables for each port monitor it starts:

- PMTAG
- ISTATE

This variable is set to a unique port monitor tag by the SAC. The port monitor uses this tag to identify itself in response to **sac** messages. ISTATE is used to indicate to the port monitor what its initial internal state should be. ISTATE is set to "enabled" or "disabled" to indicate that the port monitor is to start in the enabled or disabled state respectively.

Message Formats

The SAC performs a periodic sanity poll of the port monitors. The SAC communicates with port monitors through FIFOs. A port monitor should open `_pmpipe`, in the current directory, to receive messages from the SAC and `../_sacpipe` to send return messages to the SAC.

This section describes the messages that may be sent from the SAC to a port monitor (**sac** messages), and from a port monitor to the SAC (port monitor messages). These messages are sent through FIFOs and are in the form of C structures.

sac Messages

The format of messages from the SAC is defined by the structure **sacmsg**:

```
struct sacmsg
{
    int sc_size; /* size of optional data portion */
    char sc_type; /* type of message */
};
```

The SAC may send four types of messages to port monitors. The type of message is indicated by setting the **sc_type** field of the **sacmsg** structure to one of the following:

```
SC_STATUS          status request
SC_ENABLE          enable message
SC_DISABLE         disable message
SC_READDB         message indicating that the port
                  monitor's _pmtab file should be read
```

The **sc_size** field indicates the size of the optional data part of the message. See "Message Classes." For Solaris, **sc_size** should always be set to 0. A port monitor must respond to every message sent by the sac.

Port Monitor Messages

The format of messages from a port monitor to the SAC is defined by the structure **pmmsg**:

```
struct pmmsg {
    char pm_type; /* type of message */
    unchar pm_state; /* current state of port monitor */
    char pm_maxclass; /* maximum message class this port
                     monitor understands */
    char pm_tag[PMTAGSIZE + 1]; /* port monitor's tag */
    int pm_size; /* size of optional data portion */
};
```

Port monitors may send two types of messages to the SAC. The type of message is indicated by setting the **pm_type** field of the **pmmsg** structure to one of the following:

```
PM_STATUS    state information
```

PM_UNKNOWN negative acknowledgment

For both types of messages, the **pm_tag** field is set to the port monitor's tag and the **pm_state** field is set to the port monitor's current state. Valid states are:

PM_STARTING starting

PM_ENABLED enabled

PM_DISABLED disabled

PM_STOPPING stopping

The current state reflects any changes caused by the last message from the SAC. The status message is the normal return message. The negative acknowledgment should be sent only when the message received is not understood. **pm_size** indicates the size of the optional data part of the message. **pm_maxclass** is used to specify a message class. Both are discussed under "Message Classes." In Solaris, always set **pm_maxclass** to 1 and **sc_size** to 0. Port monitors may never initiate messages; they may only respond to messages that they receive.

Message Classes

The concept of message class has been included to accommodate possible SAF extensions. The messages described above are all class 1 messages. None of these messages contains a variable data portion; all pertinent information is contained in the message header. If new messages are added to the protocol, they will be defined as new message classes (for example, class 2). The first message the SAC sends to a port monitor will always be a class 1 message. Since all port monitors, by definition, understand class 1 messages, the first message the SAC sends is guaranteed to be understood. In its response to the SAC, the port monitor sets the **pm_maxclass** field to the maximum message class number for that port monitor. The SAC will not send messages to a port monitor from a class with a larger number than the value of **pm_maxclass**. Requests that require messages of a higher class than the port monitor can understand will fail. For Solaris, always set **pm_maxclass** to 1.

Note: For any given port monitor, messages of class **pm_maxclass** and messages of all classes with values lower than **pm_maxclass** are valid. Thus, if the **pm_maxclass** field is set to 3, the port monitor understands messages of classes 1, 2, and 3. Port monitors may not generate messages; they may only respond to messages. A port monitor's response must be of the same class as the originating message. Since only the SAC can generate messages, this protocol will function even if the port monitor is capable of dealing with messages of a higher class than the SAC can generate. **pm_size** (an element of the **pmsg** structure) and **sc_size** (an element of the **sacmsg** structure) indicate the size of the optional data part of the message. The format of this part of the message is undefined. Its definition is inherent in the type of message. For Solaris, always set both **sc_size** and **pm_size** to 0.

Administrative Interface

The SAC

Administrative File _sactab

This section discusses the port monitor administrative files available under the SAC.

The service access controller's administrative file contains information about all the port monitors for which the SAC is responsible. This file exists on the delivered system. Initially, it is empty except for a single comment line that contains the version number of the SAC. Port monitors are added to the system by making entries in the SAC's administrative file. These entries should be made using the administrative command **sacadm(1M)** with a **-a** option. **sacadm(1M)** is also used to remove entries from the SAC's administrative file. Each entry in the SAC's administrative file contains the following information.

- PMTAG** A unique tag that identifies a particular port monitor. The system administrator is responsible for naming a port monitor. This tag is then used by the SAC to identify the port monitor for all administrative purposes. PMTAG may consist of up to 14 alphanumeric characters.
- PMTYPE** The type of the port monitor. In addition to its unique tag, each port monitor has a type designator. The type designator identifies a group of port monitors that are different invocations of the same entity. `ttymon` and `listen` are examples of valid port monitor types. The type designator is used to facilitate the administration of groups of related port monitors. Without a type designator, the system administrator has no way of knowing which port monitor tags correspond to port monitors of the same type. PMTYPE may consist of up to 14 alphanumeric characters.
- FLGS** The flags that are currently defined are:
- d** When started, do not enable the port monitor.
 - x** Do not start the port monitor.
- If no flag is specified, the default action is taken. By default a port monitor is started and enabled.
- RCNT** The number of times a port monitor may fail before being placed in a failed state. Once a port monitor enters the failed state, the SAC will not try to restart it. If a count is not specified when the entry is created, this field is set to 0. A restart count of 0 indicates that the port monitor is not to be restarted when it fails.
- COMMAND**
A string representing the command that will start the port monitor. The first component of the string, the command itself, must be a full path name.

**The Port Monitor
Administrative File
_pmtab**

Each port monitor will have two directories for its exclusive use. The current directory will contain files defined by the SAF (`_pmtab`, `_pid`) and the per-service configuration scripts, if they exist. The directory `/var/saf/pmtag`, where `pmtag` is the tag of the port monitor, is available for the port monitor's private files. Each port monitor has its own administrative file. The **pmadm(1M)** command should be used to add, remove, or modify service entries in this file. Each time a change is made using **pmadm(1M)**, the corresponding port monitor rereads its administrative file. Each entry in a port monitor's administrative

file defines how the port monitor treats a specific port and what service is to be invoked on that port. Some fields must be present for all types of port monitors. Each entry must include a service tag to identify the service uniquely and an identity to be assigned to the service when it is started (for example, root).

Note: The combination of a service tag and a port monitor tag uniquely define an instance of a service. The same service tag may be used to identify a service under a different port monitor. The record must also contain port monitor specific data (for example, for a `ttymon` port monitor, this will include the prompt string which is meaningful to `ttymon`). Each type of port monitor must provide a command that takes the necessary port monitor-specific data as arguments and outputs these data in a form suitable for storage in the file. The `ttysadm(1M)` command does this for `ttymon` and `nlsadmin(1M)` does it for `listen`. For a user-defined port monitor, a similar administrative command must also be supplied. Each service entry in the port monitor administrative file must have the following format and contain the information listed below:

```
svctag:flgs:id:reserved:reserved:reserved:pmspecific# comment
```

SVCTAG is a unique tag that identifies a service. This tag is unique only for the port monitor through which the service is available. Other port monitors may offer the same or other services with the same tag. A service requires both a port monitor tag and a service tag to identify it uniquely. SVCTAG may consist of up to 14 alphanumeric characters. The service entries are defined as:

FLGS Flags with the following meanings may currently be included in this field:

- x Do not enable this port. By default the port is enabled.
- u Create a utmp entry for this service. By default no utmp entry is created for the service.

ID The identity under which the service is to be started. The identity has the form of a login name as it appears in `/etc/passwd`.

PMSPECIFIC

Examples of port monitor information are addresses, the name of a process to execute, or the name of a STREAMS pipe to pass a connection through. This information will vary to meet the needs of each different type of port monitor.

COMMENT

A comment associated with the service entry. Note: Port monitors may ignore the `u` flag if creating a utmp entry for the service is not appropriate to the manner in which the service is to be invoked. Some services may not start properly unless utmp entries have been created for them (for example, login). Each port monitor administrative file must contain one special comment of the form:

```
# VERSION=value
```

where *value* is an integer that represents the port monitor's version number. The version number defines the format of the port monitor administrative file. This comment line is created automatically when a port monitor is added to the system. It appears on a line by itself, before the service entries.

Monitor-Specific Administrative Command

Previously, two pieces of information included in the **_pmtab** file were described: the port monitor's version number and the port monitor part of the service entries in the port monitor's **_pmtab** file. When a new port monitor is added, the version number must be known so that the **_pmtab** file can be correctly initialized. When a new service is added, the port monitor part of the **_pmtab** entry must be formatted correctly. Each port monitor must have an administrative command to perform these two tasks. The person who defines the port monitor must also define such an administrative command and its input options. When the command is invoked with these options, the information required for the port monitor part of the service entry must be correctly formatted for inclusion in the port monitor's **_pmtab** file and must be written to the standard output. To request the version number the command must be invoked with a **-V** option; when it is invoked in this way, the port monitor's current version number must be written to the standard output. If the command fails for any reason during the execution of either of these tasks, no data should be written to standard output.

The Port Monitor/Service Interface

The interface between a port monitor and a service is determined solely by the service. Two mechanisms for invoking a service are presented here as examples.

New Service Invocations

The first interface is for services that are started anew with each request. This interface requires the port monitor to first **fork(2)** a child process. The child will eventually become the designated service by performing an **exec(1)**. Before the **exec(1)** happens, the port monitor may take some port monitor-specific action; however, one action that must occur is the interpretation of the per-service configuration script, if one is present. This is done by calling the library routine **doconfig(3N)**.

Standing Service Invocations

The second interface is for invocations of services that are actively running. To use this interface, a service must have one end of a stream pipe open and be prepared to receive connections through it.

Port Monitor Requirements

To implement a port monitor, several generic requirements must be met. This section summarizes these requirements. In addition to the port monitor itself, an administrative command must be supplied.

Initial Environment

When a port monitor is started, it expects an initial execution environment in which:

- It has no file descriptors open
- It cannot be a process group leader
- It has an entry in **/etc/utmp** of type **LOGIN_PROCESS**

- An environment variable, ISTATE, is set to "enabled" or "disabled" to indicate the port monitor's correct initial state
- An environment variable, PMTAG, is set to the port monitor's assigned tag
- The directory that contains the port monitor's administrative files is its current directory
- The port monitor is able to create private files in the directory `/var/saf/tag`, where *tag* is the port monitor's tag
- The port monitor is running with user id 0 (root)

Important Files

Relative to its current directory, the following key files exist for a port monitor.

- _config** The port monitor's configuration script. The port monitor configuration script is run by the SAC. The SAC is started by **init(1M)** as a result of an entry in `/etc/inittab` that calls **sac(1M)**.
- _pid** The file into which the port monitor writes its process id.
- _pmtab** The port monitor's administrative file. This file contains information about the ports and services for which the port monitor is responsible.
- _pmpipe** The FIFO through which the port monitor will receive messages from the SAC.
- svctag** The per-service configuration script for the service with the tag *svctag*.
- ../_sacpipe** The FIFO through which the port monitor will send messages to **sac(1M)**.

Port Monitor Responsibilities

A port monitor is responsible for performing the following tasks in addition to its port monitor function:

- Write its process id into the file **_pid** and place an advisory lock on the file
- Terminate gracefully on receipt of the signal SIGTERM
- Follow the protocol for message exchange with the SAC

A port monitor must perform the following tasks during service invocation:

- Create a **utmp** entry if the requested service has the **u** flag set in **_pmtab**
 Note: Port monitors may ignore this flag if creating a **utmp** entry for the service does not make sense because of the manner in which the service is to be invoked. On the other hand, some services may not start properly unless utmp entries have been created for them.
- Interpret the per-service configuration script for the requested service, if it exists, by calling the **doconfig(3N)** library routine

Configuration Files and Scripts

The library routine **doconfig**(3N), defined in **libnsl.so**, interprets the configuration scripts contained in the files **/etc/saf/_sysconfig** (the per-system configuration file), and **/etc/saf/pmtag/_config** (per-port monitor configuration files); and in **/etc/saf/pmtag/svctag** (per-service configuration files). Its syntax is:

```
#include <sac.h>
int doconfig (int fd, char *script, long rflag);
```

script is the name of the configuration script; *fd* is a file descriptor that designates the stream to which stream manipulation operations are to be applied; *rflag* is a bitmask that indicates the mode in which script is to be interpreted. *rflag* may take two values, NORUN and NOASSIGN, which may be or'd. If *rflag* is zero, all commands in the configuration script are eligible to be interpreted. If *rflag* has the NOASSIGN bit set, the assign command is considered illegal and will generate an error return. If *rflag* has the NORUN bit set, the run and runwait commands are considered illegal and will generate error returns. If a command in the script fails, the interpretation of the script ceases at that point and a positive integer is returned; this number indicates which line in the script failed. If a system error occurs, a value of -1 is returned. If a script fails, the process whose environment was being established should not be started. In the example, **doconfig**(3N) is used to interpret a per-service configuration script.

```
...
    if ((i = doconfig (fd, svctag, 0)) != 0){
        error ("doconfig failed on line %d of script %s",i,svctag);
    }
}
```

The Per-System Configuration File

The per-system configuration file, **/etc/saf/_sysconfig**, is delivered empty. It may be used to customize the environment for all services on the system by writing a command script in the interpreted language described in this chapter and on the **doconfig**(3N) manpage. When the SAC is started, it calls the **doconfig**(3N) function to interpret the per-system configuration script. The SAC is started when the system enters multiuser mode.

Per-Port Monitor Configuration Files

Per-port monitor configuration scripts (**/etc/saf/pmtag/_config**) are optional. They allow the user to customize the environment for any given port monitor and for the services that are available through the ports for which that port monitor is responsible. Per-port monitor configuration scripts are written in the same language used for per-system configuration scripts. The per-port monitor configuration script is interpreted when the port monitor is started. The port monitor is started by the SAC after the SAC has itself been started and after it has run its own configuration script, **/etc/saf/_sysconfig**. The per-port monitor configuration script may override defaults provided by the per-system configuration script.

The Configuration Language

Per-Service Configuration Files

Per-service configuration files allow the user to customize the environment for a specific service. For example, a service may require special privileges that are not available to the general user. Using the language described in the **doconfig(3N)** manpage, you can write a script that will grant or limit such special privileges to a particular service offered through a particular port monitor. The per-service configuration may override defaults provided by higher-level configuration scripts. For example, the per-service configuration script may specify a set of STREAMS modules other than the default set.

The language in which configuration scripts are written consists of a sequence of commands, each of which is interpreted separately. The following reserved keywords are defined: **assign**, **push**, **pop**, **runwait**, and **run**. The comment character is #. Blank lines are not significant. No line in a command script may exceed 1024 characters.

assign *variable=value*

Used to define environment variables; *variable* is the name of the environment variable and *value* is the value to be assigned to it. The value assigned must be a string constant; no form of parameter substitution is available. *value* may be quoted. The quoting rules are those used by the shell for defining environment variables. **assign** will fail if space cannot be allocated for the new variable or if any part of the specification is invalid.

push *module1[, module2, module3, . . .]*

Used to push STREAMS modules onto the stream designated by **fd**; *module1* is the name of the first module to be pushed, *module2* is the name of the second module to be pushed, and so on. The command will fail if any of the named modules cannot be pushed. If a module cannot be pushed, the subsequent modules on the same command line will be ignored and modules that have already been pushed will be popped.

pop [*module*]

Used to pop STREAMS modules off the designated stream. If **pop** is invoked with no arguments, the top module on the stream is popped. If an argument is given, modules will be popped one at a time until the named module is at the top of the stream. If the named module is not on the designated stream, the stream is left as it was and the command fails. If *module* is the special keyword **ALL**, then all modules on the stream will be popped. Note that only modules above the top-most driver are affected.

runwait *command*

The **runwait** command runs a command and waits for it to complete; *command* is the path name of the command to be run. The command is run with **/bin/sh -c** prepended to it; shell scripts may thus be executed from configuration scripts. The **runwait** command will fail if *command* cannot be found or cannot be executed, or if *command* exits with a nonzero status.

run *command*

The **run** command is identical to **runwait** except that it does not wait for

command to complete; *command* is the path name of the command to be run. **run** will not fail unless it is unable to create a child process to execute the command. Although they are syntactically indistinguishable, some of the commands available to **run** and **runwait** are interpreter built-in commands. Interpreter built-ins are used when it is necessary to alter the state of a process within the context of that process. The **doconfig** interpreter built-in commands are similar to the shell special commands and, like these, they do not spawn another process for execution. See the **sh**(1) man page. The initial set of built-in commands is: **cd**, **ulimit**, **umask**.

Sample Port Monitor Code

This example shows an example of a "null" port monitor that simply responds to messages from the SAC.

```
# include <stdlib.h>
# include <stdio.h>
# include <unistd.h>
# include <fcntl.h>
# include <signal.h>
# include <sac.h>

char Scratch[BUFSIZ]; /* scratch buffer */
char Tag[PMTAGSIZE + 1]; /* port monitor's tag */
FILE *Fp; /* file pointer for log file */
FILE *Tfp; /* file pointer for pid file */
char State; /* portmonitor's current state*/

main(argc, argv)
    int argc;
    char *argv[];
{
    char *istate;
    strcpy(Tag, getenv("PMTAG"));
    /*
     * open up a log file in port monitor's private directory
     */
    sprintf(Scratch, "/var/saf/%s/log", Tag);
    Fp = fopen(Scratch, "a+");
    if (Fp == (FILE *)NULL)
        exit(1);
    log(Fp, "starting");
    /*
     * retrieve initial state (either "enabled" or "disabled") and set
     * State accordingly
     */
    istate = getenv("ISTATE");
```

```

    sprintf(Scratch, "ISTATE is %s", ystate);
    log(Fp, Scratch);
    if (!strcmp(ystate, "enabled"))
        State = PM_ENABLED;
    else if (!strcmp(ystate, "disabled"))
        State = PM_DISABLED;
    else {
        log(Fp, "invalid initial state");
        exit(1);
    }
    sprintf(Scratch, "PMTAG is %s", Tag);
    log(Fp, Scratch);
/*
 * set up pid file and lock it to indicate that we are active
 */
    Tfp = fopen("_pid", "w");
    if (Tfp == (FILE *)NULL) {
        log(Fp, "couldn't open pid file");
        exit(1);
    }
    if (lockf(fileno(Tfp), F_TEST, 0) < 0) {
        log(Fp, "pid file already locked");
        exit(1);
    }

    log(Fp, "locking file");
    if (lockf(fileno(Tfp), F_LOCK, 0) < 0) {
        log(Fp, "lock failed");
        exit(1);
    }
    fprintf(Tfp, "%d", getpid());
    fflush(Tfp);

/*
 * handle poll messages from the sac ... this function never returns
 */
    handlepoll();
    pause();
    fclose(Tfp);
    fclose(Fp);
}

handlepoll()
{
    int pfd; /* file descriptor for incoming pipe */

```

```

    int sfd; /* file descriptor for outgoing pipe */
    struct sacmsg sacmsg; /* incoming message */
    struct pmmsg pmmsg; /* outgoing message */
/*
 * open pipe for incoming messages from the sac
 */
    pfd = open("_pmpipe", O_RDONLY | O_NONBLOCK);
    if (pfd < 0) {
        log(Fp, "_pmpipe open failed");
        exit(1);
    }
/*
 * open pipe for outgoing messages to the sac
 */
    sfd = open("../_sacpipe", O_WRONLY);
    if (sfd < 0) {
        log(Fp, "_sacpipe open failed");
        exit(1);
    }
/*
 * start to build a return message; we only support class 1 messages
 */
    strcpy(pmmsg.pm_tag, Tag);
    pmmsg.pm_size = 0;
    pmmsg.pm_maxclass = 1;
/*
 * keep responding to messages from the sac
 */
    for (;;) {
        if (read(pfd, &sacmsg, sizeof(sacmsg)) != sizeof(sacmsg)) {
            log(Fp, "_pmpipe read failed");
            exit(1);
        }
/*
 * determine the message type and respond appropriately
 */
        switch (sacmsg.sc_type) {
            case SC_STATUS:
                log(Fp, "Got SC_STATUS message");
                pmmsg.pm_type = PM_STATUS;
                pmmsg.pm_state = State;
                break;
            case SC_ENABLE:
                /*note internal state change below*/
                log(Fp, "Got SC_ENABLE message");

```

```

        pmmsg.pm_type = PM_STATUS;
        State = PM_ENABLED;
        pmmsg.pm_state = State;
        break;
    case SC_DISABLE:
        /*note internal state change below*/
        log(Fp, "Got SC_DISABLE message");
        pmmsg.pm_type = PM_STATUS;
        State = PM_DISABLED;
        pmmsg.pm_state = State;
        break;
    case SC_READDB:
        /*
         * if this were a fully functional port
         * monitor it would read _pmtab here
         * and take appropriate action
         */
        log(Fp, "Got SC_READDB message");
        pmmsg.pm_type = PM_STATUS;
        pmmsg.pm_state = State;
        break;
    default:
        sprintf(Scratch, "Got unknown message <%d>",
            sacmsg.sc_type);
        log(Fp, Scratch);
        pmmsg.pm_type = PM_UNKNOWN;
        pmmsg.pm_state = State;
        break;
    }
}
/*
 * send back a response to the poll
 * indicating current state
 */
        if (write(sfd, &pmmsg, sizeof(pmmsg)) != sizeof(pmmsg))
            log(Fp, "sanity response failed");
    }
}
/*
 * general logging function
 */
log(fp, msg)
    FILE *fp;
    char *msg;
{
    fprintf(fp, "%d; %s\n", getpid(), msg);
}

```

```

        fflush(fp);
    }

```

The sac.h Header File

The following example shows the sac.h header file.

```

/* length in bytes of a utmp id */
# define IDLEN 4
/* wild character for utmp ids */
# define SC_WILDC 0xff
/* max len in bytes for port monitor tag */
# define PMTAGSIZE 14
/*
 * values for rflag in doconfig()
 */
/* don't allow assign operations */
# define NOASSIGN 0x1
/* don't allow run or runwait operations */
# define NORUN 0x2
/*
 * message to SAC (header only). This header is forever fixed. The
 * size field (pm_size) defines the size of the data portion of the
 * message, which follows the header. The form of this optional data
 * portion is defined strictly by the message type (pm_type).
 */
struct pmmmsg {
    char pm_type;          /* type of message */
    uchar pm_state;       /* current state of pm */
    char pm_maxclass;     /* max message class this port monitor
                           understands */
    char pm_tag[PMTAGSIZE + 1]; /* pm's tag */
    int pm_size;          /* size of opt data portion */
};
/*
 * pm_type values
 */
# define PM_STATUS 1 /* status response */
# define PM_UNKNOWN 2 /* unknown message was received */
/*
 * pm_state values
 */
/*
 * Class 1 responses
 */
# define PM_STARTING 1 /* monitor in starting state */
# define PM_ENABLED 2 /* monitor in enabled state */

```

```

# define PM_DISABLED 3 /* monitor in disabled state */
# define PM_STOPPING 4 /* monitor in stopping state */
/*
 * message to port monitor
 */
struct sacmsg {
    int sc_size; /* size of optional data portion */
    char sc_type; /* type of message */
};
/*
 * sc_type values
 * These represent commands that the SAC sends to a port monitor.
 * These commands are divided into "classes" for extensibility. Each
 * subsequent "class" is a superset of the previous "classes" plus
 * the new commands defined within that "class". The header for all
 * commands is identical; however, a command may be defined such that
 * an optional data portion may be sent in addition to the header.
 * The format of this optional data piece is self-defining based on
 * the command. Important note: the first message sent by the SAC
 * will always be a class 1 message. The port monitor response
 * indicates the maximum class that it is able to understand. Another
 * note is that port monitors should only respond to a message with
 * an equivalent class response (i.e. a class 1 command causes a
 * class 1 response).
 */
/*
 * Class 1 commands (currently, there are only class 1 commands)
 */
# define SC_STATUS 1 /* status request *
# define SC_ENABLE 2 /* enable request */
# define SC_DISABLE 3 /* disable request */
# define SC_READDB 4 /* read pmtab request */
/*
 * 'errno' values for Saferrno, note that Saferrno is used by both
 * pmdm and sacadm and these values are shared between them
 */
# define E_BADARGS 1 /* bad args/ill-formed cmd line */
# define E_NOPRIV 2 /* user not priv for operation */
# define E_SAFERR 3 /* generic SAF error */
# define E_SYSERR 4 /* system error */
# define E_NOEXIST 5 /* invalid specification */
# define E_DUP 6 /* entry already exists */
# define E_PMRUN 7 /* port monitor is running */
# define E_PMNOTRUN 8 /* port monitor is not running */
# define E_RECOVER 9 /* in recovery */

```

Directory Structure

This section gives a description of the SAF files and directories.

/etc/saf/_sysconfig

The per-system configuration script.

/etc/saf/_sactab

The SAC's administrative file. Contains information about the port monitors for which the SAC is responsible.

/etc/saf/pmtag

The home directory for port monitor *pmtag*.

/etc/saf/pmtag/_config

The per-port monitor configuration script for port monitor *pmtag*.

/etc/saf/pmtag/_pmtab Port monitor *pmtag*'s administrative file. Contains information about the services for which *pmtag* is responsible. **/etc/saf/pmtag/svctag**

The file in which the per-service configuration script for service *svctag* (available through port monitor **pmtag**) is placed. **/etc/saf/pmtag/_pid** The file in which a

port monitor writes its process id in the current directory and places an advisory lock on the file. **/etc/saf/pmtag/_pmpipe** The file in which the port monitor

receives messages from the SAC and **./_sacpipe** and sends return messages to the SAC. **/var/saf/_log** The SAC's log file. **/var/saf/pmtag** The directory for files created

by port monitor *pmtag*, for example its log file.

**LIST OF
COMMANDS**

The following administrative commands relate to SAF.

sacadm(1M)

port monitor administrative command

pmadm(1M)

service administration command

SEE ALSO

exec(1), **sh(1)**, **init(1M)**, **nlsadmin(1M)**, **pmadm(1M)**, **sac(1M)**, **sacadm(1M)**, **ttyadm(1M)**, **fork(2)**, **doconfig(3N)**

NAME	sar, sa1, sa2, sadc – system activity report package
SYNOPSIS	<pre> /usr/lib/sa/sadc [t n] [ofile] /usr/lib/sa/sa1 [t n] /usr/lib/sa/sa2 [-aAbcdgkmpqruvw] [-e time] [-f filename] [-i sec] [-s time] </pre>
AVAILABILITY	SUNWaccu
DESCRIPTION	<p>System activity data can be accessed at the special request of a user (see sar(1)) and automatically, on a routine basis, as described here. The operating system contains several counters that are incremented as various system actions occur. These include counters for CPU utilization, buffer usage, disk and tape I/O activity, TTY device activity, switching and system-call activity, file-access, queue activity, inter-process communications, and paging. For more general system statistics, use iostat(1M), sar(1), or vmstat(1M).</p> <p>See <i>Solaris 1.x to Solaris 2.x Transition Guide</i> for device naming conventions for disks. sadc and two shell procedures, sa1 and sa2, are used to sample, save, and process this data.</p> <p>sadc, the data collector, samples system data <i>n</i> times, with an interval of <i>t</i> seconds between samples, and writes in binary format to <i>ofile</i> or to standard output. The sampling interval <i>t</i> should be greater than 5 seconds; otherwise, the activity of sadc itself may affect the sample. If <i>t</i> and <i>n</i> are omitted, a special record is written. This facility can be used at system boot time, when booting to a multi-user state, to mark the time at which the counters restart from zero. For example, when accounting is enabled, the <code>/etc/init.d/perf</code> file writes the restart mark to the daily data file using the command entry:</p> <pre> su sys -c "/usr/lib/sa/sadc /var/adm/sa/sa`date +%d`" </pre> <p>The shell script sa1, a variant of sadc, is used to collect and store data in the binary file <code>/var/adm/sa/sadd</code>, where <i>dd</i> is the current day. The arguments <i>t</i> and <i>n</i> cause records to be written <i>n</i> times at an interval of <i>t</i> seconds, or once if omitted. The following entries in <code>/var/spool/cron/crontabs/sys</code> will produce records every 20 minutes during working hours and hourly otherwise:</p> <pre> 0 * * * 0-6 /usr/lib/sa/sa1 20,40 8-17 * * 1-5 /usr/lib/sa/sa1 </pre> <p>See crontab(1) for details.</p> <p>The shell script sa2, a variant of sar, writes a daily report in the file <code>/var/adm/sa/sardd</code>. See the OPTIONS section in sar(1) for an explanation of the various options. The following entry in <code>/var/spool/cron/crontabs/sys</code> will report important activities hourly during the working day:</p> <pre> 5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A </pre>

FILES **/etc/init.d/perf**
 /tmp/sa.adrfl address file
 /var/adm/sa/sadd daily data file
 /var/adm/sa/sardd daily report file
 /var/spool/cron/crontabs/sys

SEE ALSO **crontab(1), sag(1), sar(1), timex(1), iostat(1M), vmstat(1M)**
 System Administration Guide, Volume II

NAME	savecore – save a core dump of the operating system
SYNOPSIS	<code>/usr/bin/savecore [-v] directory [system-name]</code>
DESCRIPTION	<p>savecore saves a core dump of the kernel (assuming that one was made) and writes a reboot message in the shutdown log. It is meant to be called near the end of the <code>/etc/init.d/sysetup</code> file after the system boots. However, it is not normally run by default; you must edit that file to enable it.</p> <p>savecore checks the core dump to be certain it corresponds with the version of the operating system currently running. If it does, savecore saves the core image in the file <code>directory/vmcore.n</code> and the kernel's namelist in <code>directory/unix.n</code>. The trailing <code>.n</code> in the pathnames is replaced by a number which grows every time savecore is run in that directory.</p> <p>Before savecore writes out a core image, it reads a number from the file <code>directory/minfree</code>. This is the minimum number of kilobytes that must remain free on the file system containing <code>directory</code>. If there is less free space on the file system containing <code>directory</code> than the number of kilobytes specified in <code>minfree</code>, the core dump is not saved. If the <code>minfree</code> file does not exist, savecore always writes out the core file (assuming that a core dump was taken).</p> <p>savecore also logs a reboot message using facility <code>LOG_AUTH</code> (see <code>syslog(3)</code>). If the system crashed as a result of a panic, savecore logs the panic string too.</p> <p>If the core dump corresponds to a namelist other than the currently running kernel (represented by <code>/dev/ksyms</code>) that namelist must be supplied as <code>system-name</code>.</p>
OPTIONS	<code>-v</code> Verbose. Enable verbose error messages from savecore .
FILES	<p><code>directory/vmcore.n</code> <code>directory/unix.n</code> <code>directory/bounds</code> <code>directory/minfree</code> <code>/dev/ksyms</code> the kernel namelist <code>/etc/init.d/sysetup</code> <code>/var/crash/'uname -n'</code> normal core image directory</p>
SEE ALSO	<code>compress(1)</code> , <code>crash(1M)</code> , <code>syslog(3)</code>
BUGS	<p>savecore can be fooled into thinking a core dump is the wrong size.</p> <p>You must run savecore very soon after booting — before the swap space containing the crash dump is overwritten by programs currently running.</p> <p>Core images produced by SPARCstation1 systems, and from machines with discontinuous physical memory, are sparse and contain holes. For example, a core image of an 8 megabyte SPARCstation 1 might contain 3 to 4 megabytes of useful information, and thus only occupy 3 to 4 megabytes of disk space, yet contain enough holes to appear to be 36</p>

megabytes in size. However, copying the core image will manifest the holes, so that this copy will require 36 megabytes of disk space. If it is necessary to move a core image, it is strongly recommended that the core image be compressed with **compress(1)** before the transfer. The compressed image may later be uncompressed on a system with sufficient disk space.

NAME	sendmail – send mail over the internet
SYNOPSIS	<pre> /usr/lib/sendmail [-ba] [-bd] [-bi] [-bm] [-bp] [-bs] [-bt] [-bv] [-B type] [-C file] [-d X] [-F fullname] [-f name] [-h N] [-M id] [-n] [-o xvalue] [-p protocol] [-q [time]] [-q Xstring] [-r name] [-t] [-v] [-X logfile] [address ...] </pre>
AVAILABILITY	SUNWcsr
DESCRIPTION	<p>sendmail sends a message to one or more people, routing the message over whatever networks are necessary. sendmail does internetwork forwarding as necessary to deliver the message to the correct place.</p> <p>sendmail is not intended as a user interface routine; other programs provide user-friendly front ends; sendmail is used only to deliver pre-formatted messages.</p> <p>With no flags, sendmail reads its standard input up to an EOF, or a line with a single dot, and sends a copy of the letter found there to all of the addresses listed. It determines the network to use based on the syntax and contents of the addresses.</p> <p>Local addresses are looked up in the local aliases(4) file, or by using the YP name service, and aliased appropriately. In addition, if there is a .forward file in a recipient's home directory, sendmail forwards a copy of each message to the list of recipients that file contains. Aliasing can be prevented by preceding the address with a backslash. Normally the sender is not included in alias expansions, for example, if 'john' sends to 'group', and 'group' includes 'john' in the expansion, then the letter will not be delivered to 'john'.</p> <p>sendmail will also route mail directly to other known hosts in a local network. The list of hosts to which mail is directly sent is maintained in the file /usr/lib/mailhosts.</p> <p>If a letter is found to be undeliverable, it is returned to the sender with diagnostics that indicate the location and nature of the failure; or, the letter is placed in a dead.letter file in the sender's home directory.</p>
OPTIONS	<p>-ba Go into ARPANET mode. All input lines must end with a RETURN-LINEFEED, and all messages will be generated with a RETURN-LINEFEED at the end. Also, the From: and Sender: fields are examined for the name of the sender.</p> <p>-bd Run as a daemon, waiting for incoming SMTP connections.</p> <p>-bi Initialize the aliases(4) database.</p> <p>-bm Deliver mail in the usual way (default).</p> <p>-bp Print a summary of the mail queue.</p> <p>-bs Use the SMTP protocol as described in RFC 821. This flag implies all the operations of the -ba flag that are compatible with SMTP.</p> <p>-bt Run in address test mode. This mode reads addresses and shows the steps in parsing; it is used for debugging configuration tables.</p>

- bv** Verify names only — do not try to collect or deliver a message. Verify mode is normally used for validating users or mailing lists.
- B type** indicate body *type* (**7BIT** or **8BITMIME**)
- C file** Use alternate configuration file.
- d X** Set debugging value to *X*.
- F fullname** Set the full name of the sender.
- f name** Sets the name of the “from” person (that is, the sender of the mail). **-f** can only be used by “trusted” users (who are listed in the configuration file).
- h N** Set the hop count to *N*. The hop count is incremented every time the mail is processed. When it reaches a limit, the mail is returned with an error message, the victim of an aliasing loop.
- M id** Attempt to deliver the queued message with message-id *id*. This option is supported for backward compatibility and the **-qI** option is preferred.
- n** Do not do aliasing.
- o xvalue** Set option *x* to the specified *value*. **Processing Options** are described below.
- p protocol** Set the sending protocol. Programs are encouraged to set this. The *protocol* field can be in form *protocol:host* to set both the sending protocol and the sending host. For example:
- pUUCP:uunet**
- sets the sending *protocol* to **UUCP** and the sending host to **uunet**. (Some existing programs use **-oM** to set the *r* and *s* macros; this is equivalent to using **-p**).
- q[time]** Process saved messages in the queue at given intervals. If *time* is omitted, process the queue once. *time* is given as a tagged number, with **s** being seconds, **m** being minutes, **h** being hours, **d** being days, and **w** being weeks. For example, **-q1h30m** or **-q90m** would both set the timeout to one hour thirty minutes.
- q Xstring** Run the queue once, limiting the jobs to those matching *Xstring*. The key letter *X* can be:
- I** to limit based on queue identifier (see **-M** option).
R to limit based on recipient (see **-R** option).
S to limit based on sender.
- A particular queued job is accepted if one of the corresponding addresses contains the indicated *string*.
- r name** An alternate and obsolete form of the **-f** flag.

- R *string*** Go through the queue of pending mail and attempt to deliver any message with a recipient containing the specified string. This is useful for clearing out mail directed to a machine which has been down for awhile. This option is supported for backward compatibility and the **-qR** option is preferred.
- t** Read message for recipients. **To:**, **Cc:**, and **Bcc:** lines will be scanned for people to send to. The **Bcc:** line will be deleted before transmission. Any addresses in the argument list will be suppressed.
- v** Go into verbose mode. Alias expansions will be announced, etc.
- X *logfile*** Log all traffic in and out of **sendmail** in the indicated *logfile* for debugging mailer problems. This produces a lot of data very quickly and should be used sparingly.

Processing Options

There are a number of "random" options that can be set from a configuration file. Options are represented by single characters. The syntax of this line is:

Ovalue

This sets option *o* to be *value*. Depending on the option, *value* may be

- a string
- an integer
- a boolean (with legal values **t**, **T**, **f**, or **F**; the default is TRUE)
- a time interval.

The options supported are:

- aN** If set, wait up to *N* minutes for an "@:@" entry to exist in the **aliases**(4) database before starting up. If it does not appear in *N* minutes, rebuild the database (if the **D** option is also set) or issue a warning.
- Afile** Specify possible alias file(s).
- bN/M** Insist on at least *N* blocks free on the filesystem that holds the queue files before accepting email via SMTP. If there is insufficient space **sendmail** gives a 452 response to the **MAIL** command. This invites the sender to try again later. The optional *M* is a maximum message size advertised in the **ESMTP EHLO** response. It is currently otherwise unused.
- Bc** Set the blank substitution character to *c*. Unquoted spaces in addresses are replaced by this character. Defaults to SPACE (i.e., no change is made).
- c** If an outgoing mailer is marked as being expensive, don't connect immediately.
- CN** Checkpoints the queue every *N* (default **10**) addresses sent. If your system crashes during delivery to a large list, this prevents retransmission to any but the last *N* recipients.

dx	Deliver in mode <i>x</i> . Legal modes are: i Deliver interactively (synchronously) b Deliver in background (asynchronously) q Just queue the message (deliver during queue run) Defaults to b if no option is specified, i if it is specified but given no argument (i.e., Od is equivalent to Odi).
D	If set, rebuild the /etc/mail/aliases database if necessary and possible. If this option is not set, sendmail will never rebuild the aliases database unless explicitly requested using -bi , or newaliases(1) is invoked.
ex	Dispose of errors using mode <i>x</i> . The values for <i>x</i> are: p Print error messages (default) q No messages, just give exit status m Mail back errors w Write back errors (mail if user not logged in) e Mail back errors and give zero exit stat always
Efile/message	Prepend error messages with the indicated message. If it begins with a slash, it is assumed to be the pathname of a file containing a message (this is the recommended setting). Otherwise, it is a literal message. The error file might contain the name, email address, and/or phone number of a local postmaster who could provide assistance in to end users. If the option is missing or null, or if it names a file which does not exist or which is not readable, no message is printed.
f	Save Unix-style From lines at the front of headers. Normally they are assumed redundant and discarded.
Fmode	The file mode for queue files.
gn	Set the default group ID for mailers to run in to <i>n</i> . Defaults to 1 . The value can also be given as a symbolic group name.
hN	The maximum hop count. Messages that have been processed more than <i>N</i> times are assumed to be in a loop and are rejected. Defaults to 25 .
Hfile	Specify the help file for SMTP.
i	Ignore dots in incoming messages. This is always disabled (that is, dots are always accepted) when reading SMTP mail.
I	Insist that the name server be running to resolve host names. If this is not set and the name server is not running, the /etc/hosts file (see hosts(4)) will be considered complete. In general, you do want to set this option if your /etc/hosts file does not include all hosts known to you or if you are using the MX (mail forwarding) feature of the name server. The name server will still be consulted even if this option is not set, but sendmail will feel free to resort to reading /etc/hosts if the name server

	is not available. Thus, you should <i>never</i> set this option if you do not run the name server.
j	If set, send error messages in MIME format (see RFC1341 and RFC1344 for details).
Jpath	Set the path for searching for users' .forward files. The default is \$z/.forward . Some sites that use the automounter may prefer to change this to /var/forward/\$u to search a file with the same name as the user in a system directory. It can also be set to a sequence of paths separated by colons; sendmail stops at the first file it can successfully and safely open. For example, /var/forward/\$u:\$z/.forward will search first in /var/forward/username and then in ~username/.forward (but only if the first file does not exist).
kN	The maximum number of open connections that will be cached at a time. The default is one. This delays closing the current connection until either this invocation of sendmail needs to connect to another host or it terminates. Setting it to zero defaults to the old behavior, that is, connections are closed immediately.
Ktimeout	The maximum amount of time a cached connection will be permitted to idle without activity. If this time is exceeded, the connection is immediately closed. This value should be small (on the order of ten minutes). Before sendmail uses a cached connection, it always sends a NOOP (no operation) command to check the connection; if this fails, it reopens the connection. This keeps your end from failing if the other end times out. The point of this option is to be a good network neighbor and avoid using up excessive resources on the other end. The default is five minutes.
l	If there is an Errors-To: header, send error messages to the addresses listed there. They normally go to the envelope sender. Use of this option causes sendmail to violate RFC 1123.
Ln	Set the default log level to <i>n</i> . Defaults to 9 .
m	Send to me too, even if I am in an alias expansion.
Mx value	Set the macro <i>x</i> to <i>value</i> . This is intended only for use from the command line.
n	Validate the RHS of aliases when rebuilding the aliases(4) database.
o	Assume that the headers may be in old format, i.e., spaces delimit names. This actually turns on an adaptive algorithm: if any recipient address contains a comma, parenthesis, or angle bracket, it will be assumed that commas already exist. If this flag is not on, only commas delimit names. Headers are always output with commas between the names.
Options	Set server SMTP options. The options are <i>key=value</i> pairs. Known keys are:

Port Name/number of listening port (defaults to **smtp**)
Addr Address mask (defaults **INADDR_ANY**)
Family Address family (defaults to **INET**)
Listen Size of listen queue (defaults to **10**)

The **Address** mask may be a numeric address in dot notation or a network name.

p *opt,opt,...* Set the privacy *options*. “Privacy” is really a misnomer; many of these are just a way of insisting on stricter adherence to the SMTP protocol. The *options* can be selected from:

public Allow open access
needmailhelo Insist on **HELO** or **EHLO** command before **MAIL**
needexpnhelo Insist on **HELO** or **EHLO** command before **EXPN**
noexpn Disallow **EXPN** entirely
needvrfyhelo Insist on **HELO** or **EHLO** command before **VRIFY**
novrfy Disallow **VRIFY** entirely
restrictmailq Restrict **mailq** command
restrictqrun Restrict **-q** command line flag
goaway Disallow essentially all SMTP status queries
authwarnings Put X-Authentication-Warning: headers in messages

The **goaway** pseudo-flag sets all flags except **restrictmailq** and **restrictqrun**. If **mailq** is restricted, only people in the same group as the queue directory can print the queue. If queue runs are restricted, only root and the owner of the queue directory can run the queue. Authentication Warnings add warnings about various conditions that may indicate attempts to spoof the mail system, such as using a non-standard queue directory.

Ppostmaster If set, copies of error messages will be sent to the named *postmaster*. Only the header of the failed message is sent. Since most errors are user problems, this is probably not a good idea on large sites, and arguably contains all sorts of privacy violations, but it seems to be popular with certain operating systems vendors.

qfactor Use *factor* as the multiplier in the map function to decide when to just queue up jobs rather than run them. This value is divided by the difference between the current load average and the load average limit (**x** flag) to determine the maximum message priority that will be sent. Defaults to **600000**.

Qdir Use the named *dir* as the queue directory.

r *timeouts* Timeout reads after *time* interval. The *timeouts* argument is a list of *keyword=value* pairs. The recognized timeouts and their default values, and their minimum values specified in RFC 1123 section 5.3.2 are:

initial wait for initial greeting message [5m, 5m]

helo	reply to HELO or EHLO command [5m, none]
mail	reply to MAIL command [10m, 5m]
rcpt	reply to RCPT command [1h, 5m]
datainit	reply to DATA command [5m, 2m]
datablock	data block read [1h, 3m]
datafinal	reply to final "." in data [1h, 10m]
rset	reply to RSET command [5m, none]
quit	reply to QUIT command [2m, none]
misc	reply to NOOP and VERB commands [2m, none]
command	command read [1h, 5m]
ident	IDENT protocol timeout [30s, none]

All but **command** apply to client SMTP. For backward compatibility, a timeout with no *keyword=* part will set all of the longer values.

s Be super-safe when running things, i.e., always instantiate the queue file, even if you are going to attempt immediate delivery. **sendmail** always instantiates the queue file before returning control the client under any circumstances.

Sfile Log statistics in the named *file*.

ttzinfo Set the local time zone info to *tzinfo* — for example, "PST8PDT." Actually, if this is not set, the **TZ** environment variable is cleared (so the system default is used); if set but null, the user's **TZ** variable is used, and if set and non-null the **TZ** variable is set to this value.

Trtime/wtime Set the queue timeout to *rtime*. After this interval, messages that have not been successfully sent will be returned to the sender. Defaults to five days (**5d**). The optional *wtime* is the time after which a warning message is sent. If it is missing or zero then no warning messages are sent.

un Set the default userid for mailers to *n*. Mailers without the *S* flag in the mailer definition will run as this user. Defaults to **1**. The value can also be given as a symbolic user name.

v Run in verbose mode. If this is set, **sendmail** adjusts options **c** (don't connect to expensive mailers) and **d** (delivery mode) so that all mail is delivered completely in a single job so that you can see the entire delivery process. Option **v** should *never* be set in the configuration file; it is intended for command line use only.

Vfallbackhost If specified, the *fallbackhost* acts like a very low priority MX on every host. This is intended to be used by sites with poor network connectivity.

w If you are the "best" (that is, lowest preference) MX for a given host, you should normally detect this situation and treat that condition specially, by forwarding the mail to a UUCP feed, treating it as local, or whatever. However, in some cases (such as Internet firewalls) you may want to try to connect directly to that host as though it had no MX records at all.

Setting this option causes **sendmail** to try this. The downside is that errors in your configuration are likely to be diagnosed as "host unknown" or "message timed out" instead of something more meaningful. This option is disrecommended.

xLA	When the system load average exceeds <i>LA</i> , just queue messages (i.e., don't try to send them). Defaults to 8 .
XLA	When the system load average exceeds <i>LA</i> , refuse incoming SMTP connections. Defaults to 12 .
yfact	The indicated factor (<i>fact</i>) is added to the priority (thus <i>lowering</i> the priority of the job) for each recipient, i.e., this value penalizes jobs with large numbers of recipients. Defaults to 30000 .
Y	If set, deliver each job that is run from the queue in a separate process. Use this option if you are short of memory, since the default tends to consume considerable amounts of memory while the queue is being processed.
zfact	The indicated factor (<i>fact</i>) or is multiplied by the message class (determined by the Precedence: field in the user header and the P lines in the configuration file) and subtracted from the priority. Thus, messages with a higher Priority: will be favored. Defaults to 1800 .
Zfact	The <i>fact</i> or is added to the priority every time a job is processed. Thus, each time a job is processed, its priority will be decreased by the indicated value. In most environments this should be positive, since hosts that are down are all too often down for a long time. Defaults to 90000 .
7	Strip input to seven bits for compatibility with old systems. This shouldn't be necessary.

All options can be specified on the command line using the **-o** flag, but most will cause **sendmail** to relinquish its setuid permissions. The options that will not cause this are **b, d, e, E, i, L, m, o, p, r, s, v, C**, and **7**. Also, **M** (define macro) when defining the **r** or **s** macros is also considered "safe."

If the first character of the user name is a vertical bar, the rest of the user name is used as the name of a program to pipe the mail to. It may be necessary to quote the name of the user to keep **sendmail** from suppressing the blanks from between arguments.

sendmail returns an exit status describing what it did. The codes are defined in **/usr/include/sysexits.h**.

EX_OK	Successful completion on all addresses.
EX_NOUSER	User name not recognized.
EX_UNAVAILABLE	Catchall. Necessary resources were not available.
EX_SYNTAX	Syntax error in address.
EX_SOFTWARE	Internal software error, including bad arguments.
X_OSERR	Temporary operating system error, such as "cannot fork".

EX_NOHOST Host name not recognized.

EX_TEMPFAIL Message could not be sent immediately, but was queued.

If invoked as **newaliases**, **sendmail** rebuilds the alias database. If invoked as *mailq*, **sendmail** prints the contents of the mail queue.

FILES

dead.letter unmailable text

/etc/mail/sendmail.cf defines environment for **sendmail**

/var/spool/mqueue/* temp files and queued mail

~/forward list of recipients for forwarding messages

SEE ALSO

biff(1B), **mail(1)**, **mailx(1)**, **newaliases(1)**, **aliases(4)** **hosts(4)**

Su, Zaw-Sing, and Jon Postel, *The Domain Naming Convention for Internet User Applications*, RFC 819, Network Information Center, SRI International, Menlo Park, Calif., August 1982.

Postel, Jon, *Simple Mail Transfer Protocol*, RFC 821, Network Information Center, SRI International, Menlo Park, Calif., August 1982.

Crocker, Dave, *Standard for the Format of ARPA-Internet TextMessages*, RFC 822, Network Information Center, SRI International, Menlo Park, Calif., August 1982.

NAME	server_upgrade – upgrade clients of a heterogeneous OS server
SYNOPSIS	server_upgrade -d <install_image_dir> [-p <profile>]
DESCRIPTION	<p>Use this command to upgrade clients of a heterogeneous OS server that have different platforms (SPARC, x86) or platform groups (for example, sun4d, sun4L) from the OS server. This command is necessary because clients of an OS server are not upgraded during a standard upgrade if the installation image does not support their platform or platform group.</p> <p>Following are the steps for performing a standard upgrade of an OS server and upgrading clients, followed by the steps for upgrading clients with different platforms and platform groups using the server_upgrade command. The steps assume you are mounting the Solaris CD locally, remotely, or otherwise making it available.</p> <ol style="list-style-type: none"> 1. Using the Solaris CD that matches the platform of the OS server, boot the OS server and perform a standard upgrade. Only clients that have the same platform and platform group supported on the installation image are upgraded. For example, when you boot a SPARC server using a Solaris SPARC CD; all clients that are SPARC and share the same platform group are upgraded. 2. Reboot the OS server. 3. Insert a platform-specific CD into the CD-ROM drive. For example, if the OS server is a SPARC system, which shares services for both SPARC and x86 clients, you would load the x86 Solaris CD at this point. 4. At the root prompt, type: <ul style="list-style-type: none"> <li style="margin-left: 40px;">server_upgrade -d <install_image_dir> [-p <profile>] <p>The command upgrades the platform-specific services for clients on the OS server from the installation image.</p> 5. Reboot the OS server. 6. Repeat steps 3-5 to upgrade platforms or platform groups of other clients.
OPTIONS	<p>-p <profile> Specify the full path to a custom JumpStart profile (a text file that defines how to install Solaris software on a system). For information on setting up a profile, see Installing Solaris Software.</p> <p style="margin-left: 40px;">NOTE: The profile must have the keyword set to upgrade.</p> <p>-d <install_image_dir> Specify the path to the installation image. For example, /cdrom/cdrom0.</p>
EXAMPLES	The following example shows the states of an OS server and its clients each time the server_upgrade command is used to upgrade clients. The scenario uses the server_upgrade command once to upgrade clients with different platforms (x86), and once to upgrade a client with a different platform group (sun4L).

Initial State of Solaris

The OS server is a sparc.sun4d, running Solaris 2.4, sharing the following services:

- > Solaris 2.4 for sparc.{sun4c, sun4d, sun4e, sun4m, and sun4L}
- > Solaris 2.4 for i386.i86pc
- > Solaris 2.3 for sparc.{sun4c, sun4d, sun4e, sun4m}

Initial client states are:

Host name	Is A ...	Running ...
red	sparc.sun4c	Solaris 2.4
blue	sparc.sun4e	Solaris 2.4
yellow	sparc.sun4L	Solaris 2.4
green	i386.i86pc	Solaris 2.4
purple	sparc.sun4c	Solaris 2.3
brown	sparc.sun4e	Solaris 2.3

Upgrade the 2.4 OS server to 2.5

Use the SPARC Solaris 2.5 CD to upgrade the Solaris 2.4 OS server, then reboot the OS server.

After rebooting the OS server, it is running Solaris 2.5 and sharing the following services:

- > Solaris 2.5 for sparc.{sun4c, sun4d, sun4m}
- > Solaris 2.3 for sparc.{sun4c, sun4d, sun4e, sun4m}

Client states are:

Host name	Is A ...	Running ...	And is Now...
* red	sparc.sun4c	Solaris 2.5	bootable
blue	sparc.sun4e	Solaris 2.4	not bootable
yellow	sparc.sun4L	Solaris 2.4	not bootable
green	i386.i86pc	Solaris 2.4	not bootable
* purple	sparc.sun4c	Solaris 2.5	bootable
* brown	sparc.sun4e	Solaris 2.3	bootable

NOTE: Client brown can still be booted because it is running Solaris 2.3, which is supported by the OS server, and because Solaris 2.3 supports sun4e.

Upgrade the x86 clients and services

Insert the x86 Solaris 2.5 CD and type:

```
server_upgrade -d /cdrom/cdrom0
```

After rebooting the OS server, it is running Solaris 2.5 and sharing the following services:

- > Solaris 2.5 for sparc.{sun4c, sun4d, sun4m}
- > Solaris 2.5 for i386.i86pc
- > Solaris 2.3 for sparc.{sun4c, sun4d, sun4e, sun4m}

Client states are:

Host name	Is A ...	Running ...	And is Now...
red	sparc.sun4c	Solaris 2.5	bootable
blue	sparc.sun4e	Solaris 2.4	not bootable

yellow	sparc.sun4L	Solaris 2.4	not bootable
* green	i386.i86pc	Solaris 2.5	bootable
purple	sparc.sun4c	Solaris 2.5	bootable
brown	sparc.sun4e	Solaris 2.3	bootable

**Upgrade the sun4L
(Hardware Partner)
client and services**

Insert the Hardware Edition Solaris 2.5 CD and type:

server_upgrade -d /cdrom/cdrom0

After rebooting the OS server, it is running Solaris 2.5 and sharing the following services:

```
> Solaris 2.5 for sparc.{sun4c, sun4d, sun4m, sun4L}
> Solaris 2.5 for i386.i86pc
> Solaris 2.3 for sparc.{sun4c, sun4d, sun4e, sun4m}
```

Client states are:

Host name	Is A ...	Running ...	And is Now...
red	sparc.sun4c	Solaris 2.5	bootable
blue	sparc.sun4e	Solaris 2.4	not bootable
* yellow	sparc.sun4L	Solaris 2.5	bootable
green	i386.i86pc	Solaris 2.5	bootable
purple	sparc.sun4c	Solaris 2.5	bootable
brown	sparc.sun4e	Solaris 2.3	bootable

State of sun4e clients

Client blue is not bootable because sun4e systems are not supported by Solaris 2.5. However, it can be made bootable again by using the Solstice Host Manager and adding the Solaris 2.4 services to the OS server.

Client states are:

Host name	Is A ...	Running ...	And is Now...
red	sparc.sun4c	Solaris 2.5	bootable
* blue	sparc.sun4e	Solaris 2.4	not bootable
yellow	sparc.sun4L	Solaris 2.5	bootable
green	i386.i86pc	Solaris 2.5	bootable
purple	sparc.sun4c	Solaris 2.5	bootable
brown	sparc.sun4e	Solaris 2.3	bootable

NAME	setmnt – establish mount table
SYNOPSIS	/usr/sbin/setmnt
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>setmnt creates the /etc/mnttab table which is needed for both the mount and umount commands. setmnt reads standard input and creates a mnttab entry for each line. Input lines have the format:</p> <p style="text-align: center;"><i>filesystem node</i></p> <p>where <i>filesystem</i> is the name of the file system's "special file" (such as /dev/dsk/c?d?s?) and <i>node</i> is the root name of that file system. Thus <i>filesystem</i> and <i>node</i> become the first two strings in the mount table entry.</p>
FILES	/etc/mnttab
SEE ALSO	mount(1M)
BUGS	Problems may occur if <i>filesystem</i> or <i>node</i> are longer than 32 characters. setmnt silently enforces an upper limit on the maximum number of mnttab entries.

NAME	setuname – changes machine information
SYNOPSIS	setuname [-t][-n <i>node</i>][-s <i>name</i>]
DESCRIPTION	<p>setuname changes the parameter value for the system name and node name. Each parameter can be changed using setuname and the appropriate option.</p> <p>Either or both the -s and -n options must be given when invoking setuname.</p> <p>The system architecture may place requirements on the size of the system and network node name. The command will issue a fatal warning message and an error message if the name entered is incompatible with the system requirements.</p>
OPTIONS	<p>-t Temporary change. No attempt will be made to create a permanent change.</p> <p>-n <i>node</i> Changes the node name. <i>node</i> specifies the new network node name and can consist of alphanumeric characters and the special characters dash, underbar, and dollar sign.</p> <p>-s <i>name</i> Changes the system name. <i>name</i> specifies new system name and can consist of alphanumeric characters and the special characters dash, underbar, and dollar sign.</p>
NOTES	<p>setuname attempts to change the parameter values in two places: the running kernel and, as necessary per implementation, to cross system reboots. A temporary change changes only the running kernel.</p>

NAME	share – make local resource available for mounting by remote systems
SYNOPSIS	share [-F <i>FSType</i>] [-o <i>specific_options</i>] [-d <i>description</i>] [<i>pathname</i>]
AVAILABILITY	SUNWcsu
DESCRIPTION	The share command exports, or makes a resource available for mounting, through a remote file system of type <i>FSType</i> . If the option -F <i>FSType</i> is omitted, the first file system type listed in <i>/etc/dfs/fstypes</i> is used as default. For a description of NFS specific options, see share_nfs (1M). <i>pathname</i> is the pathname of the directory to be shared. When invoked with no arguments, share displays all shared file systems.
OPTIONS	<p>-F <i>FSType</i> Specify the filesystem type.</p> <p>-o <i>specific_options</i> The <i>specific_options</i> are used to control access of the shared resource. (See share_nfs(1M) for the NFS specific options.) They may be any of the following:</p> <p> rw <i>pathname</i> is shared read/write to all clients. This is also the default behavior.</p> <p> rw=client[:client]... <i>pathname</i> is shared read/write only to the listed clients. No other systems can access <i>pathname</i>.</p> <p> ro <i>pathname</i> is shared read-only to all clients.</p> <p> ro=client[:client]... <i>pathname</i> is shared read-only only to the listed clients. No other systems can access <i>pathname</i>.</p> <p>-d <i>description</i> The -d flag may be used to provide a description of the resource being shared.</p>
EXAMPLES	This line will share the <i>/disk</i> file system read-only at boot time. <pre>share -F nfs -o ro /disk</pre>
FILES	<p><i>/etc/dfs/dfstab</i> list of share commands to be executed at boot time</p> <p><i>/etc/dfs/fstypes</i> list of file system types, NFS by default</p> <p><i>/etc/dfs/sharetab</i> system record of shared file systems</p>
SEE ALSO	mountd (1M), nfsd (1M), share_nfs (1M), shareall (1M), unshare (1M)
NOTES	<p>Export (old terminology): file system sharing used to be called exporting on SunOS 4.x, so the share command used to be invoked as exportfs(1B) or <i>/usr/sbin/exportfs</i>.</p> <p>If share commands are invoked multiple times on the same filesystem, the last share invocation supersedes the previous—the options set by the last share command replace the old options. For example, if read-write permission was given to usera on <i>/somefs</i>, then to give read-write permission also to userb on <i>/somefs</i>:</p> <pre>example% share -F nfs -o rw=usera:userb /somefs</pre>

This behavior is not limited to sharing the root filesystem, but applies to all filesystems.

NAME	share_nfs – make local NFS filesystems available for mounting by remote systems
SYNOPSIS	share [-F <i>nfs</i>] [-o <i>specific_options</i>] [-d <i>description</i>] <i>pathname</i>
DESCRIPTION	The share command makes local filesystems available for mounting by remote systems. If no argument is specified, then share displays all filesystems currently shared, including NFS filesystems and filesystems shared through other distributed file system packages.
OPTIONS	<p>-o <i>specific_options</i></p> <p>Specify options in a comma-separated list of keywords and attribute-value-assertions for interpretation by the file-system-type-specific command.</p> <p>If <i>specific_options</i> is not specified, then by default sharing will be read-write to all clients.</p> <p><i>specific_options</i> can be any combination of the following:</p> <p>rw Sharing will be read-write to all clients.</p> <p>rw=client[:client]...</p> <p>Sharing will be read-write to the listed clients; overrides the ro suboption for the clients specified. Netgroup names may be used in place of client names unless the list is used to override an ro option.</p> <p>ro Sharing will be read-only to all clients.</p> <p>ro=client[:client]...</p> <p>Sharing will be read-only to the listed clients; overrides the rw suboption for the clients specified. Netgroup names may be used in place of client names unless the list is used to override an rw option.</p> <p>anon=uid</p> <p>Set <i>uid</i> to be the effective user ID of unauthenticated users (AUTH_DES or AUTH_KERB authentication), or to be root if AUTH_UNIX authentication is used. By default, unknown users are given the effective user ID UID_NOBODY. If <i>uid</i> is set to -1, access is denied.</p> <p>root=host[:host]...</p> <p>Only root users from the specified hosts will have root access. By default, no host has root access.</p> <p>secure Clients must use the AUTH_DES authentication of RPC to be authenticated. AUTH_UNIX authentication is the default. See the anon=uid option (above) for information about how unauthenticated requests are handled.</p> <p>nosuid By default, clients are allowed to create files on the shared file system with the setuid or setgid mode enabled. Specifying nosuid causes the server file system to silently ignore any attempt to enable the setuid or setgid mode bits.</p>

aclok Allows the NFS server to do access control for NFS Version 2 clients (running SunOS 2.4 or earlier). When **aclok** is set on the server, maximal access is given to all clients. For example, with **aclok** set, if anyone has read permissions, then everyone does. If **aclok** is not set, minimal access is given to all clients.

kerberos

Clients must use the **AUTH_KERB** authentication of RPC to be authenticated. **AUTH_UNIX** authentication is the default. See the **anon=uid** option (above) for information about how unauthenticated requests are handled.

-d description Provide a comment that describes the filesystem to be shared.

pathname Specify the pathname of the filesystem to be shared.

FILES

/etc/dfs/fstypes list of system types, NFS by default
/etc/dfs/sharetab system record of shared file systems

SEE ALSO

unshare(1M), **share(1M)**

NOTES

The command will fail if both **ro** and **rw** are specified for the same client name. If the same client name exists in both the **ro=** and **rw=** lists, the **rw** will override the **ro**, giving read/write access to the client specified.

ro=, **rw=**, and **root=** are guaranteed to work over UDP but may not work over other transport providers.

If a filesystem is shared with an **ro=** list and a **root=** list, any host that is on the **root=** list will be given only read-only access, regardless of whether that host is specified in the **ro=** list, unless **rw** is declared as the default, or the host is mentioned in an **rw=** list. The same is true if the filesystem is shared with **ro** as the default. For example, the following **share** commands will give read-only permissions to **hostb**:

```
share -F nfs -o ro=hosta,root=hostb /var
```

```
share -F nfs -o ro,root=hostb /var
```

The following will give read/write permissions to **hostb**:

```
share -F nfs -o ro=hosta,rw=hostb,root=hostb /var
```

```
share -F nfs -o root=hostb /var
```

If the filesystem being shared is a symbolic link to a valid pathname, the canonical path (the path which the symbolic link follows) will be shared.

For example, if **/export/foo** is a symbolic link to **/export/bar** (**/export/foo -> /export/bar**), the following **share** command will result in **/export/bar** as the shared pathname (and not **/export/foo**).

```
example# share -F nfs /export/foo
```

Note that an NFS mount of **server:/export/foo** will result in **server:/export/bar** really being mounted.

This line will share the **/disk** file system read-only at boot time.

share -F nfs -o ro /disk

Note that the same command entered from the command line will not share the **/disk** file system unless there is at least one file system entry in the **/etc/dfs/dfstab** file. The **mountd**(1M) and **nfsd**(1M) daemons only run if there is a file system entry in **/etc/dfs/dfstab** when starting or rebooting the system.

NAME	shareall, unshareall – share, unshare multiple resources
SYNOPSIS	shareall [-F <i>FSType</i> [, <i>FSType</i> ...]] [- <i>file</i>] unshareall [-F <i>FSType</i> [, <i>FSType</i> ...]]
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>When used with no arguments, shareall shares all resources from <i>file</i>, which contains a list of share command lines. If the operand is a hyphen (-), then the share command lines are obtained from the standard input. Otherwise, if neither a <i>file</i> nor a hyphen is specified, then the file <i>/etc/dfs/dfstab</i> is used as the default.</p> <p>Resources may be shared by specific file system types by specifying the file systems in a comma-separated list as an argument to -F.</p> <p>unshareall unshares all currently shared resources. Without a -F flag, it unshares resources for all distributed file system types.</p>
OPTIONS	-F <i>FSType</i> Specify file system type. Defaults to the first entry in <i>/etc/dfs/fstypes</i> .
FILES	<i>/etc/dfs/dfstab</i>
SEE ALSO	share (1M), unshare (1M)

NAME	showmount – show all remote mounts
SYNOPSIS	<code>/usr/sbin/showmount [-ade] [hostname]</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	showmount lists all the clients that have remotely mounted a filesystem from <i>host</i> . This information is maintained by the mountd (1M) server on <i>host</i> , and is saved across crashes in the file <code>/etc/rmtab</code> . The default value for <i>host</i> is the value returned by hostname (1).
OPTIONS	-a Print all remote mounts in the format: <i>hostname</i> : directory where <i>hostname</i> is the name of the client, and directory is the root of the file system that has been mounted. -d List directories that have been remotely mounted by clients. -e Print the list of shared file systems.
FILES	<code>/etc/rmtab</code>
SEE ALSO	hostname (1), mountd (1M) <i>SPARC: Installing Solaris Software</i> <i>x86: Installing Solaris Software</i>
BUGS	If a client crashes, its entry will not be removed from the list of remote mounts on the server.

NAME	showrev – show machine and software revision information
SYNOPSIS	<code>/usr/bin/showrev [-a] [-p] [-w] [-c <i>command</i>] [-s <i>hostname</i>]</code>
AVAILABILITY	SUNWadmc
DESCRIPTION	<p>showrev displays revision information for the current hardware and software. With no arguments, showrev shows the system revision information including hostname, hostid, release, kernel architecture, application architecture, hardware provider, domain, and kernel version.</p> <p>If a <i>command</i> is supplied with the -c option, showrev shows the PATH and LD_LIBRARY_PATH and finds out all the directories within the PATH that contain it. For each file found, its file type, revision, permissions, library information, and checksum are printed as well.</p>
OPTIONS	<p>-a Print all system revision information available. Window system and patch information are added.</p> <p>-p Print only the revision information about patches.</p> <p>-w Print only the OpenWindows revision information.</p> <p>-c <i>command</i> Print the revision information about <i>command</i>.</p> <p>-s <i>hostname</i> Perform this operation on the specified <i>hostname</i>. The -s operation completes correctly only when <i>hostname</i> is running Solaris 2.5 or later.</p>
OUTPUT	<p>Varies, based on flags passed. If no flags are passed, output similar to the following appears:</p> <pre> Hostname: system1 Hostid: 7233808e Release: 5.4 Kernel architecture: sun4m Application architecture: sparc Hardware provider: Sun_Microsystems Domain: a.network.COM Kernel version: SunOS 5.4 generic July 1994 </pre>
EXIT STATUS	<p>The following error values are returned:</p> <p>0 Successful completion.</p> <p>>0 An error occurred.</p>
SEE ALSO	<code>arch(1)</code> , <code>ldd(1)</code> , <code>mcs(1)</code> , <code>sum(1)</code>
BUGS	For the -s option to work when <i>hostname</i> is running a version of Solaris prior to 2.5, the Solstice AdminSuite must be installed on <i>hostname</i> .

NAME	shutdown – shut down system, change system state
SYNOPSIS	<code>/usr/sbin/shutdown [-y] [-g <i>grace-period</i>] [-i <i>init-state</i>] [<i>message</i>]</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>shutdown is executed by the super-user to change the state of the machine. In most cases, it is used to change from the multi-user state (state 2) to another state.</p> <p>By default, shutdown brings the system to a state where only the console has access to the operating system. This state is called single-user.</p> <p>Before starting to shut down daemons and killing processes, shutdown sends a warning message and, by default, a final message asking for confirmation. <i>message</i> is a string that is sent out following the standard warning message "The system will be shut down in ...". If the string contains more than one word, it should be contained within single (') or double (") quotation marks.</p> <p>The warning message and the user provided <i>message</i> are output when there are 7200, 3600, 1800, 1200, 600, 300, 120, 60, and 30 seconds remaining before shutdown begins. See EXAMPLES.</p> <p>System state definitions are:</p> <p>state 0 Stop the operating system.</p> <p>state 1 State 1 is referred to as the administrative state. In state 1 file systems required for multi-user operations are mounted, and logins requiring access to multi-user file systems can be used. When the system comes up from firmware mode into state 1, only the console is active and other multi-user (state 2) services are unavailable. Note that not all user processes are stopped when transitioning from multi-user state to state 1.</p> <p>state s, S State s (or S) is referred to as the single-user state. All user processes are stopped on transitions to this state. In the single-user state, file systems required for multi-user logins are unmounted and the system can only be accessed through the console. Logins requiring access to multi-user file systems cannot be used.</p> <p>state 5 Shut the machine down so that it is safe to remove the power. Have the machine remove power, if possible. The rc0 procedure is called to perform this task.</p> <p>state 6 Stop the operating system and reboot to the state defined by the initdefault entry in <code>/etc/inittab</code>. The rc6 procedure is called to perform this task.</p>
OPTIONS	<p>-y Pre-answer the confirmation question so the command can be run without user intervention.</p>

- g** *grace-period* Allow the super-user to change the number of seconds from the 60-second default.
- i** *init-state* If there are warnings, *init-state* specifies the state **init** is to be in. By default, system state $\lambda q s r q$ is used.

EXAMPLES

In the following example, **shutdown** is being executed on host **foo** and is scheduled in 120 seconds. The warning message is output 2 minutes, 1 minute, and 30 seconds before the final confirmation message.

```
example# shutdown -i S -g 120 "===== disk replacement ====="
```

```
Shutdown started. Tue Jun 7 14:51:40 PDT 1994
```

```
Broadcast Message from root (pts/1) on foo Tue Jun 7 14:51:41. . .
```

```
The system will be shut down in 2 minutes
```

```
===== disk replacement =====
```

```
Broadcast Message from root (pts/1) on foo Tue Jun 7 14:52:41. . .
```

```
The system will be shut down in 1 minutes
```

```
===== disk replacement =====
```

```
Broadcast Message from root (pts/1) on foo Tue Jun 7 14:53:41. . .
```

```
The system will be shut down in 30 seconds
```

```
===== disk replacement =====
```

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

FILES /etc/inittab controls process dispatching by **init**

SEE ALSO **boot(1M)**, **halt(1M)**, **init(1M)**, **killall(1M)**, **reboot(1M)**, **ufsdump(1M)**, **init.d(4)**, **inittab(4)**

NAME	snoop – capture and inspect network packets
SYNOPSIS	snoop [-aPDSvVNC] [-d device] [-s snaplen] [-c maxcount] [-i filename] [-o filename] [-n filename] [-t [r a d]] [-p first [, last]] [-x offset [, length]] [<i>expression</i>]
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>snoop captures packets from the network and displays their contents. snoop uses both the network packet filter and streams buffer modules to provide efficient capture of packets from the network. Captured packets can be displayed as they are received, or saved to a file for later inspection.</p> <p>snoop can display packets in a single-line summary form or in verbose multi-line forms. In summary form, only the data pertaining to the highest level protocol is displayed. For example, an NFS packet will have only NFS information displayed. The underlying RPC, UDP, IP, and ethernet frame information is suppressed but can be displayed if either of the verbose options are chosen.</p>
OPTIONS	<p>-a Listen to packets on /dev/audio (warning: can be noisy).</p> <p>-P Capture packets in non-promiscuous mode. Only broadcast, multicast, or packets addressed to the host machine will be seen.</p> <p>-d device Receive packets from the network using the interface specified by <i>device</i>. Usually le0 or ie0. The program netstat(1M), when invoked with the -i flag, lists all the interfaces that a machine has. Normally, snoop will automatically choose the first non-loopback interface it finds.</p> <p>-s snaplen Truncate each packet after <i>snaplen</i> bytes. Usually the whole packet is captured. This option is useful if only certain packet header information is required. The packet truncation is done within the kernel giving better utilization of the streams packet buffer. This means less chance of dropped packets due to buffer overflow during periods of high traffic. It also saves disk space when capturing large traces to a capture file. To capture only IP headers (no options) use a <i>snaplen</i> of 34. For UDP use 42, and for TCP use 54. You can capture RPC headers with a <i>snaplen</i> of 80 bytes. NFS headers can be captured in 120 bytes.</p> <p>-c maxcount Quit after capturing <i>maxcount</i> packets. Otherwise keep capturing until there is no disk left or until interrupted with CTRL-C.</p> <p>-i filename Display packets previously captured in <i>filename</i>. Without this option, snoop reads packets from the network interface. If a <i>filename.names</i> file is present, it is automatically loaded into snoop's IP address-to-name mapping table (See -N flag below).</p>

- o filename** Save captured packets in *filename* as they are captured. During packet capture, a count of the number of packets saved in the file is displayed. If you wish just to count packets without saving to a file, name the file **/dev/null**.
- n filename** Use *filename* as an IP address-to-name mapping table. This file must have the same format as the **/etc/hosts** file (IP address followed by the hostname).
- D** Display number of packets dropped during capture on the summary line.
- S** Display size of the entire ethernet frame in bytes on the summary line.
- t [r | a | d]** Time-stamp presentation. Time-stamps are accurate to within 4 microseconds. The default is for times to be presented in **d** (delta) format (the time since receiving the previous packet). Option **a** (absolute) gives wall-clock time. Option **r** (relative) gives time relative to the first packet displayed. This can be used with the **-p** option to display time relative to any selected packet.
- v** Verbose mode. Print packet headers in lots of detail. This display consumes many lines per packet and should be used only on selected packets.
- V** Verbose summary mode. This is halfway between summary mode and verbose mode in degree of verbosity. Instead of displaying just the summary line for the highest level protocol in a packet, it displays a summary line for each protocol layer in the packet. For instance, for an NFS packet it will display a line each for the ETHER, IP, UDP, RPC and NFS layers. Verbose summary mode output may be easily piped through **grep** to extract packets of interest. For example to view only RPC summary lines:

example# snoop -i rpc.cap -V | grep RPC
- p first [, last]** Select one or more packets to be displayed from a capture file. The *first* packet in the file is packet #1.
- x offset [, length]** Display packet data in hexadecimal and ASCII format. The *offset* and *length* values select a portion of the packet to be displayed. To display the whole packet, use an *offset* of 0. If a *length* value is not provided, the rest of the packet is displayed.
- N** Create an IP address-to-name file from a capture file. This must be set together with the **-i** option that names a capture file. The address-to-name file has the same name as the capture file with **.names** appended. This file records the IP address to hostname mapping at the capture site

- and increases the portability of the capture file. Generate a **.names** file if the capture file is to be analyzed elsewhere. Packets are not displayed when this flag is used.
- C** List the code generated from the filter expression for either the kernel packet filter, or **snoop**'s own filter.
- expression* Select packets either from the network or from a capture file. Only packets for which the expression is true will be selected. If no expression is provided it is assumed to be true.
- Given a filter expression, **snoop** generates code for either the kernel packet filter or for its own internal filter. If capturing packets with the network interface, code for the kernel packet filter is generated. This filter is implemented as a streams module, upstream of the buffer module. The buffer module accumulates packets until it becomes full and passes the packets on to **snoop**. The kernel packet filter is very efficient, since it rejects unwanted packets in the kernel before they reach the packet buffer or **snoop**. The kernel packet filter has some limitations in its implementation — it is possible to construct filter expressions that it cannot handle. In this event, **snoop** generates code for its own filter. The **-C** flag can be used to view generated code for either the kernel's or **snoop**'s own packet filter. If packets are read from a capture file using the **-i** option, only **snoop**'s packet filter is used.
- A filter *expression* consists of a series of one or more boolean primitives that may be combined with boolean operators (AND, OR, and NOT). Normal precedence rules for boolean operators apply. Order of evaluation of these operators may be controlled with parentheses. Since parentheses and other filter expression characters are known to the shell, it is often necessary to enclose the the filter expression in quotes. The primitives are:
- host** *hostname*
- True if the source or destination address is that of *hostname*. The keyword **host** may be omitted if the name does not conflict with the name of another expression primitive e.g. "**pinky**" selects packets transmitted to or received from the host **pinky** whereas "**pinky and dinky**" selects packets exchanged between hosts **pinky** AND **dinky**. Normally the IP address is used. With the **ether** qualifier the ethernet address is used, for instance, "**ether pinky**".
- ipaddr* or *etheraddr*
- Literal addresses, both IP dotted and ethernet colon are recognized. For example, "**129.144.40.13**" matches all packets with that IP address as source or destination, and similarly, "**8:0:20:f:b1:51**" matches all packets with the ethernet address as source or destination. An ethernet address beginning with a letter is interpreted as a hostname. To avoid this, prepend a zero

when specifying the address. For example, if the ethernet address is "aa:0:45:23:52:44", then specify it by add a leading zero to make it "0aa:0:45:23:52:44".

from or src

A qualifier that modifies the following **host**, **net**, **ipaddr**, **etheraddr**, **port** or **rpc** primitive to match just the source address, port, or RPC reply.

to or dst

A qualifier that modifies the following **host**, **net**, **ipaddr**, **etheraddr**, **port** or **rpc** primitive to match just the destination address, port, or RPC call.

ether A qualifier that modifies the following **host** primitive to resolve a name to an ethernet address. Normally, IP address matching is performed.

ethertype *number*

True if the ethernet type field has value *number*.
Equivalent to "**ether[12:2] = number**".

ip, arp, rarp

True if the packet is of the appropriate ethertype.

broadcast

True if the packet is a broadcast packet.
Equivalent to "**ether[2:4] = 0xffffffff**".

multicast

True if the packet is a multicast packet.
Equivalent to "**ether[0] & 1 = 1**".

apple True if the packet is an Apple Ethertalk packet.
Equivalent to "**ethertype 0x809b or ethertype 0x803f**".

decnet True if the packet is a DECNET packet.

greater *length*

True if the packet is longer than *length*.

less *length*

True if the packet is shorter than *length*.

udp, tcp, icmp

True if the IP protocol is of the appropriate type.

net *net* True if either the IP source or destination address has a network number of *net*. The **from** or **to** qualifier may be used to select packets for which the network number occurs only in the source or destination address.

port *port*

True if either the source or destination port is *port*. The *port* may be either a port number or name from `/etc/services`. The **tcp** or

udp primitives may be used to select TCP or UDP ports only. The **from** or **to** qualifier may be used to select packets for which the *port* occurs only as the source or destination.

rpc prog [, *vers* [, *proc*]]

True if the packet is an RPC call or reply packet for the protocol identified by *prog*. The *prog* may be either the name of an RPC protocol from */etc/rpc* or a program number. The *vers* and *proc* may be used to further qualify the program *version* and *procedure* number, for example, "**rpc nfs,2,0**" selects all calls and replies for the NFS null procedure. The **to** or **from** qualifier may be used to select either call or reply packets only.

gateway host

True if the packet used *host* as a gateway, that is, the ethernet source or destination address was for *host* but not the IP address.

Equivalent to "**ether host host** and not **host host**".

nofrag True if the packet is unfragmented or is the first in a series of IP fragments.

Equivalent to "**ip[6:2] & 0x1fff = 0**".

expr relop expr

True if the relation holds, where *relop* is one of >, <, >=, <=, =, !=, and *expr* is an arithmetic expression composed of numbers, packet field selectors, the **length** primitive, and arithmetic operators +, -, *, &, |, ^, and %. The arithmetic operators within *expr* are evaluated before the relational operator and normal precedence rules apply between the arithmetic operators, such as multiplication before addition. Parentheses may be used to control the order of evaluation. To use the value of a field in the packet use the following syntax:

base[*expr* [: *size*]]

where *expr* evaluates the value of an offset into the packet from a *base* offset which may be **ether**, **ip**, **udp**, **tcp**, or **icmp**. The *size* value specifies the size of the field. If not given, 1 is assumed. Other legal values are 2 and 4.

Examples:

"**ether[0] & 1 = 1**" is equivalent to **multicast**.

"**ether[2:4] = 0xffffffff**" is equivalent to **broadcast**.

"**ip[ip[0] & 0xf * 4 : 2] = 2049**" is equivalent to "**udp[0:2] = 2049**".

"**ip[0] & 0xf > 5**" selects IP packets with options.

"**ip[6:2] & 0x1fff = 0**" eliminates IP fragments.

"**udp and ip[6:2]&0x1fff = 0 and udp[6:2] != 0**" finds all packets with UDP checksums.

The **length** primitive may be used to obtain the length of the packet. For instance "**length > 60**" is equivalent to "**greater 60**", and "**ether[length - 1]**" obtains the value of the last byte in a packet.

and Perform a logical AND operation between two boolean values. The AND operation is implied by the juxtaposition of two boolean expressions, for example "**dinky pinky**" is the same as "**dinky AND pinky**".

or or , Perform a logical OR operation between two boolean values. A comma may be used instead, for example, "**dinky,pinky**" is the same as "**dinky OR pinky**".

not or ! Perform a logical NOT operation on the following boolean value. This operator is evaluated before AND or OR .

EXAMPLES

Capture all packets and display them as they are received:

```
example#snoop
```

Capture packets with host **funky** as either the source or destination and display them as they are received:

```
example#snoopfunky
```

Capture packets between **funky** and **pinky** and save them to a file. Then inspect the packets using times (in seconds) relative to the first captured packet:

```
example# snoop -o cap funky pinky
```

```
example$ snoop -i cap -t r | more
```

Look at selected packets in another capture file:

```
example$ snoop -i pkts -p99,108
```

```

99 0.0027 boutique -> sunroof  NFS C GETATTR FH=8E6C
100 0.0046 sunroof -> boutique  NFS R GETATTR OK
101 0.0080 boutique -> sunroof  NFS C RENAME FH=8E6C MTr00192 to .nfs08
102 0.0102 marmot -> viper      NFS C LOOKUP FH=561E screen.r.13.i386
103 0.0072 viper -> marmot     NFS R LOOKUP No such file or directory
104 0.0085 bugbomb -> sunroof   RLOGIN C PORT=1023 h
105 0.0005 kandinsky -> sparky  RSTAT C Get Statistics
106 0.0004 beeblebrox -> sunroof  NFS C GETATTR FH=0307
107 0.0021 sparky -> kandinsky   RSTAT R
108 0.0073 office -> jeremiah    NFS C READ FH=2584 at 40960 for 8192

```

Packet 101 Looks interesting. Take a look in more detail:

```
example$ snoop -i pkts -v -p101
```

```
ETHER: ----- Ether Header -----
ETHER:
ETHER: Packet 101 arrived at 16:09:53.59
ETHER: Packet size = 210 bytes
ETHER: Destination = 8:0:20:1:3d:94, Sun
ETHER: Source    = 8:0:69:1:5f:e, Silicon Graphics
ETHER: Ethertype = 0800 (IP)
ETHER:
IP: ----- IP Header -----
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = 00
IP:  ..0. .... = routine
IP:   ...0 .... = normal delay
IP:    .... 0... = normal throughput
IP:     .... .0.. = normal reliability
IP: Total length = 196 bytes
IP: Identification 19846
IP: Flags = 0X
IP: .0. .... = may fragment
IP: ..0. .... = more fragments
IP: Fragment offset = 0 bytes
IP: Time to live = 255 seconds/hops
IP: Protocol = 17 (UDP)
IP: Header checksum = 18DC
IP: Source address = 129.144.40.222, boutique
IP: Destination address = 129.144.40.200, sunroof
IP:
UDP: ----- UDP Header -----
UDP:
UDP: Source port = 1023
UDP: Destination port = 2049 (Sun RPC)
UDP: Length = 176
UDP: Checksum = 0
UDP:
RPC: ----- SUN RPC Header -----
RPC:
RPC: Transaction id = 665905
RPC: Type = 0 (Call)
RPC: RPC version = 2
RPC: Program = 100003 (NFS), version = 2, procedure = 1
RPC: Credentials: Flavor = 1 (Unix), len = 32 bytes
RPC:  Time = 06-Mar-90 07:26:58
RPC:  Hostname = boutique
RPC:  Uid = 0, Gid = 1
RPC:  Groups = 1
RPC: Verifier  : Flavor = 0 (None), len = 0 bytes
RPC:
NFS: ----- SUN NFS -----
NFS:
NFS: Proc = 11 (Rename)
NFS: File handle = 000016430000000100080000305A1C47
NFS:           597A0000000800002046314AFC450000
NFS: File name = MTra00192
NFS: File handle = 000016430000000100080000305A1C47
```

```
NFS:          597A000000800002046314AFC450000
NFS: File name = .nfs08
NFS:
```

View just the NFS packets between **sunroof** and **boutique**:

```
example$ snoop -i pkts rpc nfs and sunroof and boutique
```

```
 1 0.0000 boutique -> sunroof  NFS C GETATTR FH=8E6C
 2 0.0046 sunroof -> boutique NFS R GETATTR OK
 3 0.0080 boutique -> sunroof NFS C RENAME FH=8E6C MTr00192 to .nfs08
```

Save these packets to a new capture file:

```
$ snoop -i pkts -o pkts.nfs rpc nfs sunroof boutique
```

EXIT STATUS

Unless **snoop** receives an error signal, its Exit Status is zero. All abnormal exits return **1**.

SEE ALSO

netstat(1M), **bufmod**(7M), **dlpi**(7P), **ie**(7D), **le**(7D), **pfmod**(7M)

WARNINGS

The processing overhead is much higher for realtime packet interpretation. Consequently, the packet drop count may be higher. For more reliable capture, output raw packets to a file using the **-o** option and analyze the packets off-line.

Unfiltered packet capture imposes a heavy processing load on the host computer—particularly if the captured packets are interpreted realtime. This processing load further increases if verbose options are used. Since heavy use of **snoop** may deny computing resources to other processes, it should not be used on production servers. Heavy use of **snoop** should be restricted to a dedicated computer.

snoop does not reassemble IP fragments. Interpretation of higher level protocol halts at the end of the first IP fragment.

snoop may generate extra packets as a side-effect of its use. For example it may use a network name service (NIS or NIS+) to convert IP addresses to host names for display. Capturing into a file for later display can be used to postpone the address-to-name mapping until after the capture session is complete. Capturing into an NFS-mounted file may also generate extra packets.

Setting the **snaplen** (**-s** option) to small values may remove header information required for packet interpretation for higher level protocols. For complete NFS interpretation do not set **snaplen** less than 120 bytes.

snoop requires information from an RPC request to fully interpret an RPC reply. If an RPC reply in a capture file or packet range does not have a request preceding it, then only the RPC reply header will be displayed.

NOTES

snoop requires an interactive interface.

NAME	spray – spray packets
SYNOPSIS	<code>/usr/sbin/spray [-c <i>count</i>] [-d <i>delay</i>] [-l <i>length</i>] [-t <i>nettype</i>] <i>host</i></code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>spray sends a one-way stream of packets to <i>host</i> using RPC, and reports how many were received, as well as the transfer rate. The <i>host</i> argument can be either a name or an Internet address.</p> <p>spray is not useful as a networking benchmark as it uses unreliable connectionless transports, (udp for example). spray can report a large number of packets dropped when the drops were caused by spray sending packets faster than they can be buffered locally (before the packets get to the network medium).</p>
OPTIONS	<p><code>-c <i>count</i></code> Specify how many packets to send. The default value of <i>count</i> is the number of packets required to make the total stream size 100000 bytes.</p> <p><code>-d <i>delay</i></code> Specify how many microseconds to pause between sending each packet. The default is 0.</p> <p><code>-l <i>length</i></code> The <i>length</i> parameter is the numbers of bytes in the Ethernet packet that holds the RPC call message. Since the data is encoded using XDR, and XDR only deals with 32 bit quantities, not all values of <i>length</i> are possible, and spray rounds up to the nearest possible value. When <i>length</i> is greater than 1514, then the RPC call can no longer be encapsulated in one Ethernet packet, so the <i>length</i> field no longer has a simple correspondence to Ethernet packet size. The default value of <i>length</i> is 86 bytes (the size of the RPC and UDP headers).</p> <p><code>-t <i>nettype</i></code> Specify class of transports. Defaults to netpath. See rpc(3N) for a description of supported classes.</p>
SEE ALSO	rpc(3N)

NAME	ssaadm – Administration program for the SPARCstorage Array
SYNOPSIS	ssaadm [-v] <i>subcommand</i> [<i>option</i> ...] <i>pathname</i> ...
AVAILABILITY	SUNWssaop
DESCRIPTION	<p>The ssaadm program is an administrative command that manages the SPARCstorage Array. ssaadm performs a variety of control and query tasks depending on the command line arguments and options used.</p> <p>The command line must contain a subcommand and at least one pathname. It may also contain options and other parameters depending on the subcommand. The subcommand is applied to each of the pathnames on the command line.</p> <p><i>pathname</i> specifies the SPARCstorage Array controller or a disk in the SPARCstorage Array. The controller name is specified by its physical name, e.g.,</p> <p style="padding-left: 40px;">/devices/.../.../SUNW,soc@3,0/SUNW,pln@xxxxxxxx,xxxxxxxx:ctrl,</p> <p>or by a name of the form</p> <p style="padding-left: 40px;">cN,</p> <p>where <i>N</i> is the logical controller number. ssaadm uses the cN name to find an entry in the /dev/rdisk directory of a disk that is attached to the SPARCstorage Array controller. The /dev/rdisk entry is then used to determine the physical name of the SPARCstorage Array controller. A disk in the SPARCstorage Array is specified by its logical or physical device name, e.g.,</p> <p style="padding-left: 40px;">/dev/rdisk/c1t0d0s2</p> <p>or</p> <p style="padding-left: 40px;">/devices/.../.../SUNW,soc@3,0/SUNW,pln@xxxxxxxx,xxxxxxxx/ssd@0,0:c,raw.</p> <p>See disks(1M) for more information on logical names for disks and controllers.</p>
OPTIONS	<p>-v verbose mode.</p> <p>Other options specific to each subcommand are described below.</p>
USAGE Subcommands	<p>display [-p] <i>pathname</i> ...</p> <p>Display configuration information for the specified unit(s) or display performance information for the specified SPARCstorage Array controller. If <i>pathname</i> specifies the controller, the configuration information is displayed for all disks in the SPARCstorage Array.</p> <p>-p Display performance information for the specified SPARCstorage Array controller. The accumulation of the performance statistics must be enabled using the perf_statistics subcommand before displaying the performance information. If it is not, all of the I/O's per second will be displayed as zeros.</p>

The performance display reports the following information:

- BUSY** how busy the controller in the SPARCstorage Array is, expressed as a percentage.
 - IOPS** the total I/O's per second for the SPARCstorage Array.
- entries for each disk
the total number of I/O's per second.

download *-f filename pathname*

Download the prom image specified by *filename* to the SPARCstorage Array controller FEPROMs. When the download is complete the SPARCstorage Array must be reset in order to use the downloaded code. **NOTE:** The download modifies the FEPROM on the SPARCstorage Array and should be used with **CAUTION**.

fast_write *[-s] -c pathname*

fast_write *[-s] -d pathname*

fast_write *[-s] -e pathname*

Enable or disable the use of the NVRAM to enhance the performance of writes in the SPARCstorage Array. *pathname* may refer to the SPARCstorage Array controller or to an individual disk.

- s** This option causes the SPARCstorage Array to save the change so it will persist across power-cycles.
- c** Enable fast writes for synchronous writes only.
- e** Enable fast writes.
- d** Disable fast writes.

perf_statistics *-e pathname*

perf_statistics *-d pathname*

Enable or disable the accumulation of performance statistics for the specified SPARCstorage Array controller. The accumulation of performance statistics must be enabled before using the **display -p** subcommand. This subcommand can be issued only to the SPARCstorage Array controller.

- e** Enable the accumulation of performance statistics.
- d** Disable the accumulation of performance statistics.

purge *pathname*

Purge any fast write data from NVRAM for one disk, or all disks if the controller is specified. This option should be used with caution, usually only when a drive has failed.

reserve *pathname*

Reserve the specified controller(s) or disk(s) for exclusive use by the issuing host.

release *pathname*

Release a reservation held on the specified controller(s) or disk(s).

start [-t *tray-number*] *pathname*

Spin up the specified disk(s). If *pathname* specifies the controller, this action applies to all disks in the SPARCstorage Array.

-t Spin up all disks in the tray specified by *tray-number*. *pathname* must be of the controller.

stop [-t *tray-number*] *pathname*

Spin down the specified disk(s). If *pathname* specifies the controller, this action applies to all disks in the SPARCstorage Array.

-t Spin down all disks in the tray specified by *tray-number*. *pathname* must be of the controller.

sync_cache *pathname*

Flush all outstanding writes for the specified disk(s) from NVRAM to the media. If *pathname* specifies the controller, this action applies to all disks in the SPARCstorage Array.

EXIT STATUS

The following exit values are returned:

0 Successful completion.

-1 An error occurred.

SEE ALSO

disks(1M)

NOTES

For more information, see the *SPARCstorage Array Users Guide*.

NAME	ssacli – SPARCstorage Array Command Line Interface
SYNOPSIS	ssacli [<i>option</i> . . .] <i>action object</i> . . .
AVAILABILITY	SUNWssaop
DESCRIPTION	<p>The ssacli command is now obsolete and has been replaced with the ssaadm command. The ssacli command will not be available in future releases.</p> <p>The ssacli command is a command line interface application that manages the SPARCstorage Array. ssacli performs a variety of control and query tasks depending on the command line arguments and options used.</p> <p>The command line must contain an <i>action</i>, and one or in some cases more than one <i>object</i>. It may also contain options and other parameters depending on the action. The action is applied to each of the objects on the command line.</p>
OPTIONS	<p>-v verbose mode.</p> <p>Other options specific to the action are described below.</p>
USAGE	<p>In the following descriptions, <i><ctlr></i> is used as the pathname of the controller and <i><unit></i> is used as the pathname of a disk.</p> <p>The pathname of the controller can be the physical path of the controller. An example of a physical path is:</p> <p style="padding-left: 40px;">/devices/../../SUNW,soc@3,0/SUNW,plnxxxxxxxx,xxxxxxxx:ctlr</p> <p>or it can be an entry of the form</p> <p style="padding-left: 40px;">cN</p> <p>where <i>cN</i> is the logical controller number (See disks(1M) for more information on logical names for disks and controllers).</p> <p>The ssacli program uses the <i>cN</i> entry to find an entry in the /dev/rdsk directory of a disk that is attached to the SPARCstorage Array controller. The /dev/rdsk entry is then used to find the physical name of the SPARCstorage Array controller.</p> <p>The <i><unit></i> is addressed by its logical or physical address.</p> <p>An example of a logical disk address is: /dev/rdsk/c1t0d0s2.</p>
ACTIONS	<p>[-p] display <ctlr><unit> . . . Display configuration or performance information.</p> <p>-p used to display performance information. If it is not already enabled the accumulation of the performance statistics should be enabled before displaying the performance information. If it is not all of the I/O's per second will be displayed as zeros.</p> <p>The performance display shows the following information:</p> <p>BUSY in percent, how busy the MicroSparc controller in the</p>

SPARCstorage Array is

IOPS the total I/O's per second for the SPARCstorage Array entries for each disk
the total I/O's per second for that disk.

-s -f filename download <ctrl>

The SPARCstorage Array controllers prom image specified by *filename* will be down-loaded to the SPARCstorage Array controller and written into its FEPROMs.

NOTE: The **download** modifies the FEPROM on the SPARCstorage Array and should be used with **CAUTION**. When the download is complete the SPARCstorage Array must be reset in order to use the downloaded code.

-s specifies to save the downloaded image. Downloading an image without this option is not supported.

[-s] **-c fast_write <ctrl/unit>**

[-s] **-d fast_write <ctrl/unit>**

[-s] **-e fast_write <ctrl/unit>**

Select fast write mode for all units or for a specified unit.

-c enables fast writes for synchronous writes only

-d disables all fast writes

-e enables fast writes for all writes.

-s will cause the SSA to save the change so it will persist across power-cycles.

-d perf_statistics <ctrl>

-e perf_statistics <ctrl>

Disable/enable the accumulation of performance statistics.

purge <ctrl/unit>

Purges any fast write data from NVRAM for one unit, or all units if the controller is specified. This option should be used with caution, usually only when a drive has failed.

reserve <ctrl/unit>

Reserve the controller or a unit for exclusive use by the issuing host.

release <ctrl/unit>

Release a reservation.

start <ctrl/unit>...

Spin up all units in the controller or a specific unit.

-t <tray number> start <ctrl>

Spin up all units in the selected tray.

stop <ctrl/unit>...

Spin down all units in the controller or a specific unit.

-t *<tray number>* **stop** *<ctrl>*

Spin down all units in the selected tray.

sync_cache *<ctrl/unit>*

Flushes all outstanding writes for all units or for the specified unit from NVRAM to the media.

EXIT STATUS

The following exit values are returned:

0 Successful completion.

-1 An error occurred.

SEE ALSO

disks(1M), **ssaadm(1M)**

NOTES

For more information, see the *SPARCstorage Array Users Guide*.

NAME	statd – network status monitor
SYNOPSIS	<code>/usr/lib/nfs/statd</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>statd is an intermediate version of the status monitor. It interacts with lockd(1M) to provide the crash and recovery functions for the locking services on NFS. statd keeps track of the clients with processes which hold locks on a server. When the server reboots after a crash, statd sends a message to the statd on each client indicating that the server has rebooted. The client statd processes then inform the lockd on the client that the server has rebooted. The client lockd then attempts to reclaim the lock(s) from the server.</p> <p>statd on the client host also informs the statd on the server(s) holding locks for the client when the client has rebooted. In this case, the statd on the server informs its lockd that all locks held by the rebooting client should be released, allowing other processes to lock those files.</p>
FILES	<p><code>/var/statmon/sm</code> lists hosts to be contacted after a reboot <code>/var/statmon/sm.bak</code> lists hosts that could not be contacted after last reboot <code>/var/statmon/state</code> includes a number which changes during a reboot <code>/usr/include/rpcsvc/sm_inter.x</code> contains the rpcgen source code for the interface services provided by the statd daemon.</p>
SEE ALSO	lockd (1M)
NOTES	The crash of a server is only detected upon its recovery.

NAME	strace – print STREAMS trace messages
SYNOPSIS	strace [<i>mid sid level</i>] ...
AVAILABILITY	SUNWesu
DESCRIPTION	<p>strace without arguments writes all STREAMS event trace messages from all drivers and modules to its standard output. These messages are obtained from the STREAMS log driver (see log(7D)). If arguments are provided, they must be in triplets of the form <i>mid, sid, level</i>, where <i>mid</i> is a STREAMS module ID number, <i>sid</i> is a sub-ID number, and <i>level</i> is a tracing priority level. Each triplet indicates that tracing messages are to be received from the given module/driver, sub-ID (usually indicating minor device), and priority level equal to, or less than the given level. The token all may be used for any member to indicate no restriction for that attribute.</p> <p>The format of each trace message output is:</p> <pre><seq> <time> <ticks> <level> <flags> <mid> <sid> <text></pre> <ul style="list-style-type: none"> <seq> trace sequence number <time> time of message in <i>hh:mm:ss</i> <ticks> time of message in machine ticks since boot <level> tracing priority level <flags> E : message is also in the error log F : indicates a fatal error N : mail was sent to the system administrator (hardcoded as root) <mid> module ID number of source <sid> sub-ID number of source <text> formatted text of the trace message <p>Once initiated, strace will continue to execute until terminated by the user.</p>
EXAMPLES	<p>Output all trace messages from the module or driver whose module ID is 41:</p> <pre>strace 41 all all</pre> <p>Output those trace messages from driver/module ID 41 with sub-IDs 0, 1, or 2:</p> <pre>strace 41 0 1 41 1 1 41 2 0</pre> <p>Messages from sub-IDs 0 and 1 must have a tracing level less than or equal to 1. Those from sub-ID 2 must have a tracing level of 0.</p>
SEE ALSO	<p>log(7D) <i>STREAMS Programming Guide</i></p>

NOTES

- There is no restriction to the number of **strace** processes opening the STREAMS log driver at a time.
- The log-driver records the list of the triplets specified in the command invocation, and compares each potential trace message against this list to decide if it should be formatted and sent up to the **strace** process. Hence, long lists of triplets will have a greater impact on overall STREAMS performance. Running **strace** will have the most impact on the timing of the modules and drivers generating the trace messages that are sent to the **strace** process. If trace messages are generated faster than the **strace** process can handle them, some of the messages will be lost. This last case can be determined by examining the sequence numbers on the trace messages output.

NAME	strclean – STREAMS error logger cleanup program
SYNOPSIS	strclean [-a <i>age</i>] [-d <i>logdir</i>]
AVAILABILITY	SUNWcsu
DESCRIPTION	strclean is used to clean up the STREAMS error logger directory on a regular basis (for example, by using cron . By default, all files with names matching error.* in /var/adm/streams that have not been modified in the last three days are removed.
OPTIONS	-a <i>age</i> The maximum age in days for a log file can be changed using the -a option. -d <i>logdir</i> A directory other than /var/adm/streams can be specified using the -d option.
EXAMPLES	This example has the same result as running strclean with no arguments: example% strclean -d /var/adm/streams -a 3
FILES	/var/adm/streams/error.*
SEE ALSO	cron(1M) , strerr(1M) <i>STREAMS Programming Guide</i>
NOTES	strclean is typically run from cron on a daily or weekly basis.

NAME	strerr – STREAMS error logger daemon
SYNOPSIS	strerr
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>strerr receives error log messages from the STREAMS log driver (see log(7D)) and appends them to a log file. The resultant error log files reside in the directory /var/adm/streams, and are named error.mm-dd, where <i>mm</i> is the month and <i>dd</i> is the day of the messages contained in each log file.</p> <p>The format of an error log message is:</p> <pre><seq> <time> <ticks> <flags> <mid> <sid> <text></pre> <ul style="list-style-type: none"> <seq> error sequence number <time> time of message in hh:mm:ss <ticks> time of message in machine ticks since boot priority level <flags> <ul style="list-style-type: none"> T : the message was also sent to a tracing process F : indicates a fatal error N : send mail to the system administrator (hardcoded as root) <mid> module ID number of source <sid> sub-ID number of source <text> formatted text of the error message <p>Messages that appear in the error log are intended to report exceptional conditions that require the attention of the system administrator. Those messages which indicate the total failure of a STREAMS driver or module should have the F flag set. Those messages requiring the immediate attention of the administrator will have the N flag set, which causes the error logger to send the message to the system administrator using mail. The priority level usually has no meaning in the error log but will have meaning if the message is also sent to a tracer process.</p> <p>Once initiated, strerr continues to execute until terminated by the user. It is commonly executed asynchronously.</p>
FILES	/var/adm/streams/error.mm-dd error log file.
SEE ALSO	<p>log(7D)</p> <p><i>STREAMS Programming Guide</i></p>

NOTES

- There is no restriction to the number of **strerr** processes opening the STREAMS log driver at a time.
- If a module or driver is generating a large number of error messages, running the error logger will cause a degradation in STREAMS performance. If a large burst of messages are generated in a short time, the log driver may not be able to deliver some of the messages. This situation is indicated by gaps in the sequence numbering of the messages in the log files.

NAME	sttydefs – maintain line settings and hunt sequences for TTY ports
SYNOPSIS	<pre> /usr/sbin/sttydefs -a ttylabel [-b] [-f final-flags] [-i initial-flags] [-n nextlabel] /usr/sbin/sttydefs -l [ttylabel] /usr/sbin/sttydefs -r ttylabel </pre>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>sttydefs is an administrative command that maintains the line settings and hunt sequences for the system's TTY ports by making entries in, and deleting entries from the <code>/etc/ttydefs</code> file.</p> <p>sttydefs with a <code>-a</code> or <code>-r</code> option may be invoked only by the super-user. sttydefs with <code>-l</code> may be invoked by any user on the system.</p>
OPTIONS	<p><code>-a ttylabel</code> Add a record to the ttydefs file, using <i>ttylabel</i> as its label. The following describes the effect of the <code>-b</code>, <code>-n</code>, <code>-i</code>, or <code>-f</code> options when used in conjunction with the <code>-a</code> option:</p> <p><code>-b</code> Enable autobaud. Autobaud allows the system to set the line speed of a given TTY port to the line speed of the device connected to the port without the user's intervention.</p> <p><code>-f final-flags</code> Specify the value to be used in the <i>final-flags</i> field in <code>/etc/ttydefs</code>. <i>final-flags</i> must be in a format recognized by the stty command. <i>final-flags</i> are the termio(7I) settings used by ttymon after receiving a successful connection request and immediately before invoking the service on the port. If this option is not specified, sttydefs will set <i>final-flags</i> equal to the termio(7I) flags 9600 and sane.</p> <p><code>-i initial-flags</code> Specify the value to be used in the <i>initial-flags</i> field in <code>/etc/ttydefs</code>. <i>initial-flags</i> must be in a format recognized by the stty command. These flags are used by ttymon when searching for the correct baud rate. They are set prior to writing the prompt.</p> <p>If this option is not specified, sttydefs will set <i>initial-flags</i> equal to the termio(7I) flag 9600.</p> <p><code>-n nextlabel</code> Specify the value to be used in the <i>nextlabel</i> field in <code>/etc/ttydefs</code>. If this option is not specified, sttydefs will set <i>nextlabel</i> equal to <i>ttylabel</i>.</p> <p><code>-l</code> If a <i>ttylabel</i> is specified, sttydefs displays the record from <code>/etc/ttydefs</code> whose TTY label matches the specified <i>ttylabel</i>. If no <i>ttylabel</i> is specified, sttydefs displays the entire contents of <code>/etc/ttydefs</code>. sttydefs verifies that each entry it displays is correct and that the entry's <i>nextlabel</i> field references an existing</p> <p><code>-r ttylabel</code> Remove any record in the ttydefs file that has <i>ttylabel</i> as its label.</p>

OUTPUT

If successful, **sttydefs** will exit with a status of **0**. **sttydefs -l** will generate the requested information and send it to standard output.

EXAMPLES

The following command lists all the entries in the **ttydefs** file and prints an error message for each invalid entry that is detected.

```
example# sttydefs -l
```

The following shows a command that requests information for a single label and its output:

```
example# sttydefs -l 9600
```

```
-----  
9600:9600 hupcl erase ^h:9600 sane ixany tab3 hupcl erase ^h::4800  
-----
```

```
ttylabel:          9600  
initial flags:    9600 hupcl erase ^h  
final flags:      9600 sane ixany tab3 hupcl erase ^h  
autobaud:         no  
nextlabel:        4800
```

The following sequence of commands will add the labels **1200**, **2400**, **4800**, and **9600** and put them in a circular list:

```
sttydefs -a 1200 -n 2400 -i 1200 -f "1200 sane"  
sttydefs -a 2400 -n 4800 -i 2400 -f "2400 sane"  
sttydefs -a 4800 -n 9600 -i 4800 -f "4800 sane"  
sttydefs -a 9600 -n 1200 -i 9600 -f "9600 sane"
```

FILES

/etc/ttydefs

SEE ALSO

termio(7I)

NAME	su – become super-user or another user
SYNOPSIS	su [-] [<i>username</i> [<i>arg</i> ...]]
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>su allows one to become another user without logging off. The default user <i>name</i> is root (super-user).</p> <p>To use su, the appropriate password must be supplied (unless the invoker is already root). If the password is correct, su creates a new shell process that has the real and effective user ID, group IDs, and supplementary group list set to those of the specified <i>username</i>. The new shell will be the shell specified in the shell field of <i>username</i>'s password file entry (see passwd(4)). If no shell is specified, /usr/bin/sh is used (see sh(1)). To return to normal user ID privileges, type an EOF character (CTRL-D) to exit the new shell.</p> <p>Any additional arguments given on the command line are passed to the new shell. When using programs such as sh, an <i>arg</i> of the form -c string executes <i>string</i> using the shell and an arg of -r gives the user a restricted shell.</p> <p>If the first argument to su is ' - ' (dash), the environment will be changed to what would be expected if the user actually logged in as the specified user. This is accomplished by invoking the program used as the shell with a first argument value whose initial character is ' - ' (dash), thus simulating a login. If the first argument to su is not ' - ' (dash), the environment is passed along unchanged, with the exception of \$PATH, which is controlled by PATH and SUPATH in /etc/default/su.</p> <p>All attempts to become another user using su are logged in the log file /var/adm/sulog (see sulog(4)).</p>
EXAMPLES	<p>To become user bin while retaining your previously exported environment, execute:</p> <p style="padding-left: 40px;">example% su bin</p> <p>To become user bin but change the environment to what would be expected if bin had originally logged in, execute:</p> <p style="padding-left: 40px;">example% su - bin</p> <p>To execute <i>command</i> with the temporary environment and permissions of user bin, type:</p> <p style="padding-left: 40px;">example% su - bin -c "command args"</p>
ENVIRONMENT	<p>If any of the LC_* variables (LC_CTYPE, LC_MESSAGES, LC_TIME, LC_COLLATE, LC_NUMERIC, and LC_MONETARY) (see environ(5)) are not set in the environment, the operational behavior of su for each corresponding locale category is determined by the value of the LANG environment variable. If LC_ALL is set, its contents are used to override both the LANG and the other LC_* variables. If none of the above variables are set in the environment, the "C" (U.S. style) locale determines how su behaves.</p>

LC_CTYPE Determines how **su** handles characters. When **LC_CTYPE** is set to a valid value, **su** can display and handle text and filenames containing valid characters for that locale. **su** can display and handle Extended Unix Code (EUC) characters where any individual character can be 1, 2, or 3 bytes wide. **su** can also handle EUC characters of 1, 2, or more column widths. In the "C" locale, only characters from ISO 8859-1 are valid.

LC_MESSAGES Determines how diagnostic and informative messages are presented. This includes the language and style of the messages, and the correct form of affirmative and negative responses. In the "C" locale, the messages are presented in the default form found in the program itself (in most cases, U.S. English).

FILES

\$HOME/.profile user's login commands for sh and ksh
/etc/passwd system's password file
/etc/profile system-wide sh and ksh login commands
/var/adm/sulog log file
/etc/default/su the default parameters that reside here are:
SULOG: If defined, all attempts to **su** to another user are logged in the indicated file.
CONSOLE: If defined, all attempts to **su** to **root** are logged on the console.
PATH: Default path. (**/usr/bin:**)
SUPATH: Default path for a user invoking **su** to **root**. (**/usr/sbin:/usr/bin**)
SYSLOG: Determines whether the **syslog(3)** **LOG_AUTH** facility should be used to log all **su** attempts. **LOG_NOTICE** messages are generated for **su**'s to **root**, **LOG_INFO** messages are generated for **su**'s to other users, and **LOG_CRIT** messages are generated for failed **su** attempts.

SEE ALSO

csh(1), **env(1)**, **ksh(1)**, **login(1)**, **sh(1)**, **syslogd(1M)**, **syslog(3)**, **passwd(4)**, **profile(4)**, **sulog(4)**, **environ(5)**

NAME	sulogin – access single-user mode
SYNOPSIS	sulogin
AVAILABILITY	SUNWcsr
DESCRIPTION	sulogin is automatically invoked by init when the system is first started. It prompts the user to type the root password to enter system maintenance mode (single-user mode) or to type EOF (typically CTRL-D) for normal startup (multi-user mode). sulogin should never be directly invoked by the user.
FILES	/etc/sulogin
SEE ALSO	init(1M)

NAME	suninstall – install the Solaris environment				
SYNOPSIS	suninstall				
DESCRIPTION	<p>suninstall is a forms-based subsystem for installing the operating system.</p> <p>Note: suninstall only exists on the Solaris CD-ROM and should only be invoked from there. Refer to the installation manual for more details.</p> <p>suninstall allows installation of the operating system onto any standalone system. suninstall loads the software available on the CD-ROM. Refer to the installation manual for disk space requirements.</p> <p>To abort the installation procedure, use the interrupt character (typically CTRL-C).</p>				
USAGE	Refer to the installation manual for more information on the various menus and selections.				
FILES	<table><tr><td>/usr/etc/install</td><td>directory containing installation programs and scripts</td></tr><tr><td>/etc/install</td><td>directory containing suninstall data files</td></tr></table>	/usr/etc/install	directory containing installation programs and scripts	/etc/install	directory containing suninstall data files
/usr/etc/install	directory containing installation programs and scripts				
/etc/install	directory containing suninstall data files				
SEE ALSO	<p>pkginfo(1), pkgadd(1M)</p> <p><i>SPARC: Installing Solaris Software</i></p> <p><i>x86: Installing Solaris Software</i></p>				
NOTES	It is advisable to exit suninstall through the exit options from the suninstall menus.				

NAME	swap – swap administrative interface
SYNOPSIS	<pre> /usr/sbin/swap -a swapname [swaplow] [swaplen] /usr/sbin/swap -d swapname [swaplow] /usr/sbin/swap -l /usr/sbin/swap -s </pre>
AVAILABILITY	SUNWcsu
DESCRIPTION	swap provides a method of adding, deleting, and monitoring the system swap areas used by the memory manager.
OPTIONS	<p>-a swapname Add the specified swap area. This option can only be used by the super-user. <i>swapname</i> is the name of the swap file: for example, <code>/dev/dsk/c0t0d0s1</code> or a regular file. <i>swaplow</i> is the offset in 512-byte blocks into the file where the swap area should begin. <i>swaplen</i> is the desired length of the swap area in 512-byte blocks. The value of <i>swaplen</i> can not be less than 16. For example, if <i>n</i> blocks are specified, then (<i>n</i>-1) blocks would be the actual swap length. <i>swaplen</i> must be at least one page in length. One page of memory is equivalent to eight 512-byte blocks. The size of a page of memory can be determined by using the pagesize command. See pagesize(1). Since the first page of a swap file is automatically skipped, and a swap file needs to be at least one page in length, the minimum size should be a factor of 2 pagesize bytes. The size of a page of memory is machine dependent.</p> <p><i>swaplow</i> + <i>swaplen</i> must be less than or equal to the size of the swap file. If <i>swaplen</i> is not specified, an area will be added starting at <i>swaplow</i> and extending to the end of the designated file. If neither <i>swaplow</i> nor <i>swaplen</i> are specified, the whole file will be used except for the first page. Swap areas are normally added automatically during system startup by the <code>/sbin/swapadd</code> script. This script adds all swap areas which have been specified in the <code>/etc/vfstab</code> file; for the syntax of these specifications, see vfstab(4).</p> <p>To use an NFS or local file-system <i>swapname</i>, you should first create a file using mkfile(1M). A local file-system swap file can now be added to the running system by just running the swap -a command. For NFS mounted swap files, the server needs to export the file. Do this by performing the following steps:</p> <ol style="list-style-type: none"> 1. Add the following line to <code>/etc/dfs/dfstab</code>: <pre> share -F nfs -o rw=clientname,root=clientname path-to-swap-file </pre> 2. Run shareall(1M).

3. Have the client add the following lines to `/etc/vfstab`:

```
server:path-to-swap-file - local-path-to-swap-file nfs ---
local-path-to-swap-file -- swap ---
```
 4. Have the client run **mount**:

```
# mount local-path-to-swap-file
```
 5. The client can then run **swap -a** to add the swap space:

```
# swap -a local-path-to-swap-file
```
- d swapname** Delete the specified swap area. This option can only be used by the super-user. *swapname* is the name of the swap file: for example, `/dev/dsk/c0t0d0s1` or a regular file. *swaplow* is the offset in 512-byte blocks into the swap area to be deleted. If *swaplow* is not specified, the area will be deleted starting at the second page. When the command completes, swap blocks can no longer be allocated from this area and all swap blocks previously in use in this swap area have been moved to other swap areas.
- l** List the status of all the swap areas. The output has five columns:
- | | |
|---------------|---|
| path | The path name for the swap area. |
| dev | The major/minor device number in decimal if it is a block special device; zeroes otherwise. |
| swaplo | The <i>swaplow</i> value for the area in 512-byte blocks. |
| blocks | The <i>swapplen</i> value for the area in 512-byte blocks. |
| free | The number of 512-byte blocks in this area that are not currently allocated. |
- The list does not include swap space in the form of physical memory because this space is not associated with a particular swap area.
- If **swap -l** is run while *swapname* is in the process of being deleted (by **swap -d**), the string INDEL will appear in a sixth column of the swap stats.
- s** Print summary information about total swap space usage and availability:
- | | |
|------------------|--|
| allocated | The total amount of swap space (in 1024-byte blocks) currently allocated for use as backing store. |
| reserved | The total amount of swap space (in 1024-bytes blocks) not currently allocated, but claimed by memory mappings for possible future use. |
| used | The total amount of swap space (in 1024-byte blocks) that is either allocated or reserved. |
| available | The total swap space (in 1024-byte blocks) that is currently available for future reservation and allocation. |

These numbers include swap space from all configured swap areas as listed by the `-l` option, as well swap space in the form of physical memory.

SEE ALSO `pagesize(1)`, `mkfile(1M)`, `shareall(1M)`, `getpagesize(3C)`, `vfstab(4)`

WARNINGS No check is done to see if a swap area being added overlaps with an existing file system.

NAME	swmtool – install, upgrade, and remove software packages
SYNOPSIS	swmtool
AVAILABILITY	SUNWinst
DESCRIPTION	<p>The Software Manager, swmtool, is an application that installs and removes packages on a local system. In Solaris 2.5 and later, it invokes the admintool(1M), preselected to add or remove software. Using swmtool, you can add software from a product CD or from a hard disk to an installed system, or you can remove software from an installed system.</p> <p>Once you are logged in, you may run swmtool to examine the packages on your local system.</p> <p>Membership in the sysadmin group (gid 14) is used to restrict access to administrative tasks. Members of the sysadmin group can use swmtool to add or remove software packages. Non-members have read-only permissions (where applicable).</p> <p>Help is available by using the Help button.</p>
SEE ALSO	admintool(1M) <i>Solaris Advanced User's Guide</i> pkgadd(1M) pkgrm(1M)

NAME	sxconfig – configure contiguous memory for the SX video subsystem
SYNOPSIS	<pre> /usr/platform/<i>platform-name</i>/sbin/sxconfig -c /usr/platform/<i>platform-name</i>/sbin/sxconfig -d /usr/platform/<i>platform-name</i>/sbin/sxconfig [-f -n] [-l <i>limit</i>] [-s <i>size</i>] </pre>
AVAILABILITY	SUNWkvm
DESCRIPTION	<p>The sxconfig command configures contiguous memory parameters for exclusive use by the SX video system on the Desktop SPARCsystems with graphics option. <i>platform-name</i> can be found using the -i option of uname(1).</p> <p>After configuring the physically contiguous memory, using the various options described below, the system must be rebooted for the changes to take effect. If this command is being used to configure physically contiguous memory for the first time after the system software has been installed, then the system must be rebooted using the reconfiguration option (-r) of boot(1M).</p> <p>The amount of memory to be reserved depends upon the type of application. Applications that benefit from the availability of contiguous memory are those that are written to the XGL and XIL graphics and imaging foundation library APIs.</p> <p>The <i>Platform Notes: SPARCstation 10SX System Configuration Guide</i> provides more detailed information regarding how much memory to reserve for various types of graphics and imaging applications.</p> <p>sxconfig is supported only on Desktop SPARCsystems with SX graphics option.</p> <p>The interface, output, and command location are uncommitted and subject to change in future releases.</p>
OPTIONS	<p>The following options are supported:</p> <ul style="list-style-type: none"> -c Display the current configuration parameters in the driver configuration file. If the system was not rebooted after previously changing the configuration parameters, then the displayed values do not reflect the actual system setup. -d Restore all configuration parameters to the default values. By default, 0 megabytes of physically contiguous memory are reserved, fragmentation is not allowed, and 32 megabytes of memory are reserved for system use. -f Allow fragmentation. If no single chunk of memory of at least the requested size is found, allow the request to span multiple chunks. This flag also specifies that less than <i>size</i> megabytes of data may be reserved if there are not enough contiguous chunks available. If this flag is not specified, then the memory reserved must be exactly one chunk of the requested size for the request to succeed. -n Fragmentation not allowed.

- l *limit*** Specify that at least *limit* megabytes of total memory must remain for system use after the contiguous memory has been reserved.
- s *size*** Reserve *size* megabytes of contiguous memory for exclusive use by the **SX** video subsystem.

EXAMPLES

The following example reserves 16 megabytes of contiguous memory without fragmentation and indicates 32 megabytes of memory should remain for system use after reserving the contiguous memory:

```
example# sxconfig -s 16 -l 32
```

The following example is identical to the one described above except that fragmentation is allowed:

```
example# sxconfig -s 16 -f -l 32
```

The following example reports current configuration parameters in the driver configuration file:

```
example# sxconfig -c
```

This example restores all configuration parameters to the default values:

```
example# sxconfig -d
```

The following example disables fragmentation:

```
example# sxconfig -n
```

EXIT STATUS

sxconfig returns **0** on success, and a positive integer on failure.

- 1** Permission denied. Only root can run this command.
- 2** Configuration file `/kernel/drv/sx_cmем.conf` does not exist.
- 3** Illegal option.
- 4** Illegal combination of options.
- 5** Illegal argument for `-s` option. Should be an integer.
- 6** Illegal argument for `-l` option. Should be an integer.

FILES

/platform/platform-name/kernel/drv/sx_cmем contiguous memory device driver

/platform/platform-name/kernel/drv/sx_cmем.conf configuration file for contiguous memory driver

/etc/init.d/sxcmем contiguous memory startup script

SEE ALSO

uname(1), **boot(1M)**, **init(1M)**

Platform Notes: SPARCstation 10SX System Configuration Guide

NAME	sync – update the super block
SYNOPSIS	sync
AVAILABILITY	SUNWcsu
DESCRIPTION	sync executes the sync system primitive. If the system is to be stopped, sync must be called to insure file system integrity. It will flush all previously unwritten system buffers out to disk, thus assuring that all file modifications up to that point will be saved. See sync(2) for details.
SEE ALSO	sync(2)
NOTES	If you have done a write to a file on a remote machine in a Remote File Sharing environment, you cannot use sync to force buffers to be written out to disk on the remote machine. sync will only write local buffers to local disks.

NAME syncinit – set serial line interface operating parameters.

SYNOPSIS /usr/sbin/syncinit device [[baud_rate] | [keyword=value,...] | [single-word option]]

DESCRIPTION The **syncinit** utility allows the user to modify some of the hardware operating modes common to synchronous serial lines. This can be useful in troubleshooting a link, or necessary to the operation of a communications package.

If run without options, **syncinit** reports the options as presently set on the port. If options are specified, the new settings are reported after they have been made.

OPTIONS Options to **syncinit** normally take the form of a keyword, followed by an equal sign and a value. The exception is that a baud rate may be specified as a decimal integer by itself. Keywords must begin with the value shown in the options table, but may contain additional letters up to the equal sign. For example, **loop=** and **loopback=** are equivalent.

Recognized options are listed in the table below.

Keyword	Value	Effect
loop	yes	Set the port to operate in internal loopback mode. The receiver is electrically disconnected from the DCE receive data input and tied to the outgoing transmit data line. Transmit data is available to the DCE. The Digital Phase-Locked Loop (DPLL) may not be used as a clock source in this mode. If no other clocking options have been specified, perform the equivalent of txc=baud and rxc=baud .
	no	Disable internal loopback mode. If no other clocking options have been specified, perform the equivalent of txc=txc and rxc=rxc .
echo	yes	Set the port to operate in auto-echo mode. The transmit data output is electrically disconnected from the transmitter and tied to the receive data input. Incoming receive data is still visible. Use of this mode in combination with local loopback mode has no value, and should be rejected by the device driver. The auto-echo mode is useful to make a system become the endpoint of a remote loopback test.
	no	Disable auto-echo mode.
nrzi	yes	Set the port to operate with NRZI data encoding.
	no	Set the port to operate with NRZ data encoding.
txc	txc	Transmit clock source will be the TxC signal (pin 15).
	rxc	Transmit clock source will be the RxC signal (pin 17).
	baud	Transmit clock source will be the internal baud rate generator .
	pll	Transmit clock source will be the output of the DPLL circuit.

rx	rx	Receive clock source will be the RxC signal (pin 17).
	tx	Receive clock source will be the TxC signal (pin 15).
	baud	Receive clock source will be the internal baud rate generator .
	pll	Receive clock source will be the output of the DPLL circuit.
speed	<i>integer</i>	Set the baud rate to <i>integer</i> bits per second.

There are also several single-word options that set one or more parameters at a time:

Keyword	Equivalent to Options:
external	txc=txc rxc=rx loop=no
sender	txc=baud rxc=rx loop=no
internal	txc=pll rxc=pll loop=no
stop	speed=0

EXAMPLES

The following command sets the first CPU port to loop internally, use internal clocking and operate at 38400 baud:

```
example# syncinit zsh0 38400 loop=yes
device: /dev/zsh ppa: 0
speed=38400, loopback=yes, echo=no, nrzi=no, txc=baud, rxc=baud
```

The following command sets the same port's clocking, local loopback and baud rate settings to their default values:

```
example# syncinit zsh0 stop loop=no
device: /dev/zsh ppa: 0
speed=0, loopback=no, echo=no, nrzi=no, txc=txc, rxc=rx
```

SEE ALSO

syncloop(1M), **syncstat(1M)**, **intro(2)**, **ioctl(2)**, **zsh(7D)**

DIAGNOSTICS

device missing minor device number

The name *device* does not end in a decimal number that can be used as a minor device number.

bad speed: arg

The string *arg* that accompanied the **speed=** option could not be interpreted as a decimal integer.

Bad arg: arg

The string *arg* did not make sense as an option.

ioctl failure code = errno

An **ioctl(2)** system call failed. The meaning of the value of *errno* may be found in **intro(2)**.

WARNINGS

syncinit should not be used on an active serial link, unless needed to resolve an error condition. It should not be run casually, or if the user is unsure of the consequences of its use.

NAME	syncloop – synchronous serial loopback test program
SYNOPSIS	<code>/usr/sbin/syncloop [-cdlstv] device</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>The syncloop command performs several loopback tests that are useful in exercising the various components of a serial communications link.</p> <p>Before running a test, syncloop opens the designated port and configures it according to command line options and the specified test type. It announces the names of the devices being used to control the hardware channel, the channel number (ppa) corresponding to the <i>device</i> argument, and the parameters it has set for that channel. It then runs the loopback test in three phases.</p> <p>The first phase is to listen on the port for any activity. If no activity is seen for at least four seconds, syncloop proceeds to the next phase. Otherwise, the user is informed that the line is active and that the test cannot proceed, and the program exits.</p> <p>In the second phase, called the "first-packet" phase, syncloop attempts to send and receive one packet. The program will wait for up to four seconds for the returned packet. If no packets are seen after five attempts, the test fails with an exhorting message. If a packet is returned, the result is compared with the original. If the length and content do not match exactly, the test fails.</p> <p>The final phase, known as the "multiple-packet" phase, attempts to send many packets through the loop. Because the program has verified the integrity of the link in the first-packet phase, the test will not fail after a particular number of timeouts. If a packet is not seen after four seconds, a message is displayed. Otherwise, a count of the number of packets received is updated on the display once per second. If it becomes obvious that the test is not receiving packets during this phase, the user may wish to stop the program manually. The number and size of the packets sent during this phase is determined by default values, or by command line options. Each returned packet is compared with its original for length and content. If a mismatch is detected, the test fails. The test completes when the required number of packets have been sent, regardless of errors.</p> <p>After the multiple-packet phase has completed, the program displays a summary of the hardware event statistics for the channel that was tested. The display takes the following form:</p> <pre> CRC errors Aborts Overruns Underruns In <-Drops-> Out 0 0 0 0 0 0 </pre> <p>This is followed by an estimated line speed, which is an approximation of the bit rate of the line, based on the number of bytes sent and the actual time that it took to send them.</p>

OPTIONS

The options for **syncloop** are described in the following table:

Option	Parameter	Default	Description
-c	<i>packet_count</i>	100	Specifies the number of packets to be sent in the multiple-packet phase.
-d	<i>hex_data_byte</i>	<i>random</i>	Specifies that each packet will be filled with bytes with the value of <i>hex_data_byte</i> .
-l	<i>packet_length</i>	100	Specifies the length of each packet in bytes.
-s	<i>line_speed</i>	9600	Bit rate in bits per second.
-v			Sets verbose mode. If data errors occur, the expected and received data is displayed.
-t	<i>test_type</i>	<i>none</i>	A number, from 1 to 4, that specifies which test to perform. The values for <i>test_type</i> are as follows: <ol style="list-style-type: none"> 1 Internal loopback test. Port loopback is on. Transmit and receive clock sources are internal (baud rate generator). 2 External loopback test. Port loopback is off. Transmit and receive clock sources are internal. Requires a loopback plug suitable to the port under test. 3 External loopback test. Port loopback is off. Transmit and receive clock sources are external (modem). Requires that one of the local modem, the remote modem, or the remote system be set in a loopback configuration. 4 Test using predefined parameters. User defines hardware configuration and may select port parameters using the syncinit(1M) command.

All numeric options except **-d** are entered as decimal numbers (for example, **-s 19200**). If you do not provide the **-t test_type** option, **syncloop** prompts for it.

EXAMPLES

In the following command **syncloop** uses a packet length of 512 bytes over the first CPU port:

```
example# syncloop -l 512 zsh0
```

In response to the above command, **syncloop** prompts you for the test option you want.

The following command performs an internal loopback test on the first CPU port, using 5000 packets and a bit rate of 56Kbps:

```
example# syncloop -t 1 -s 56000 -c 5000 zsh0
```

SEE ALSO**syncinit(1M)**, **syncstat(1M)**, **zsh(7D)****DIAGNOSTICS*****device* missing minor device number**

The name *device* does not end in a decimal number that can be used as a minor device number.

invalid packet length: *nnn*

The packet length was specified to be less than zero or greater than 4096.

poll: nothing to read**poll: nothing to read or write.**

The **poll(2)** system call indicates that there is no input pending and/or that output would be blocked if attempted.

len *xxx* should be *yyy*

The packet that was sent had a length of *yyy*, but was received with a length of *xxx*.

nnn* packets lost in outbound queueing**nnn* packets lost in inbound queueing**

A discrepancy has been found between the number of packets sent by **syncloop** and the number of packets the driver counted as transmitted, or between the number counted as received and the number read by the program.

WARNINGS

To allow its tests to run properly, as well as prevent disturbance of normal operations, **syncloop** should only be run on a port that is not being used for any other purpose at that time.

NAME	syncstat – report driver statistics from a synchronous serial link.																		
SYNOPSIS	<code>/usr/sbin/syncstat [-c] device [interval]</code>																		
AVAILABILITY	SUNWcsu																		
DESCRIPTION	<p>The syncstat command reports the event statistics maintained by a synchronous serial device driver. The report may be a single snapshot of the accumulated totals, or a series of samples showing incremental changes. Prior to these it prints the device name being used to query a particular device driver, along with a number indicating the channel number (ppa) under control of that driver.</p> <p>Event statistics are maintained by a driver for each physical channel that it supports. They are initialized to zero at the time the driver module is loaded into the system, which may be either at boot time or when one of the driver's entry points is first called.</p> <p>The <i>device</i> argument is the name of the serial device as it appears in the <code>/dev</code> directory. For example, zsh0 specifies the first on-board serial device.</p> <p>The following is a breakdown of syncstat output:</p> <table border="0"> <tr> <td style="padding-right: 2em;">speed</td> <td>The line speed the device has been set to operate at. It is the user's responsibility to make this value correspond to the modem clocking speed when clocking is provided by the modem.</td> </tr> <tr> <td>ipkts</td> <td>The total number of input packets.</td> </tr> <tr> <td>opkts</td> <td>The total number of output packets.</td> </tr> <tr> <td>undrun</td> <td>The number of transmitter underrun errors.</td> </tr> <tr> <td>ovrrun</td> <td>The number of receiver overrun errors.</td> </tr> <tr> <td>abort</td> <td>The number of aborted received frames.</td> </tr> <tr> <td>crc</td> <td>The number of received frames with CRC errors.</td> </tr> <tr> <td>isize</td> <td>The average size (in bytes) of input packets.</td> </tr> <tr> <td>osize</td> <td>The average size (in bytes) of output packets.</td> </tr> </table>	speed	The line speed the device has been set to operate at. It is the user's responsibility to make this value correspond to the modem clocking speed when clocking is provided by the modem.	ipkts	The total number of input packets.	opkts	The total number of output packets.	undrun	The number of transmitter underrun errors.	ovrrun	The number of receiver overrun errors.	abort	The number of aborted received frames.	crc	The number of received frames with CRC errors.	isize	The average size (in bytes) of input packets.	osize	The average size (in bytes) of output packets.
speed	The line speed the device has been set to operate at. It is the user's responsibility to make this value correspond to the modem clocking speed when clocking is provided by the modem.																		
ipkts	The total number of input packets.																		
opkts	The total number of output packets.																		
undrun	The number of transmitter underrun errors.																		
ovrrun	The number of receiver overrun errors.																		
abort	The number of aborted received frames.																		
crc	The number of received frames with CRC errors.																		
isize	The average size (in bytes) of input packets.																		
osize	The average size (in bytes) of output packets.																		
OPTIONS	<p>-c Clear the accumulated statistics for the device specified. This may be useful when it is not desirable to unload a particular driver, or when the driver is not capable of being unloaded.</p> <p>interval syncstat samples the statistics every <i>interval</i> seconds and reports incremental changes. The output reports line utilization for input and output in place of average packet sizes. These are the relationships between bytes transferred and the baud rate, expressed as percentages. The loop repeats indefinitely, with a column heading printed every twenty lines for convenience.</p>																		

EXAMPLES

```
example# syncstat zsh0
```

speed	ipkts	opkts	undrun	ovrrun	abort	crc	isize	osize
9600	15716	17121	0	0	1	3	98	89

```
example# syncstat -c zsh0
```

speed	ipkts	opkts	undrun	ovrrun	abort	crc	isize	osize
9600	0	0	0	0	0	0	0	0

```
example# syncstat zsh0 5
```

ipkts	opkts	undrun	ovrrun	abort	crc	iutil	outil
12	10	0	0	0	0	5%	4%
22	60	0	0	0	0	3%	90%
36	14	0	0	0	1	51%	2%

In this final example a new line of output is generated every five seconds.

SEE ALSO

syncinit(1M), **syncloop(1M)**, **zsh(7D)**

DIAGNOSTICS

bad interval: *arg*

The argument *arg* is expected to be an interval and could not be understood.

device missing minor device number

The name *device* does not end in a decimal number that can be used as a minor device number.

baud rate not set

The *interval* option is being used and the baud rate on the device is zero. This would cause a divide-by-zero error when computing the line utilization statistics.

WARNINGS

Underrun, overrun, frame-abort, and CRC errors have a variety of causes. Communication protocols are typically able to handle such errors and initiate recovery of the transmission in which the error occurred. Small numbers of such errors are not a significant problem for most protocols. However, because the overhead involved in recovering from a link error can be much greater than that of normal operation, high error rates can greatly degrade overall link throughput. High error rates are often caused by problems in the link hardware, such as cables, connectors, interface electronics or telephone lines. They may also be related to excessive load on the link or the supporting system.

The percentages for input and output line utilization reported when using the *interval* option may occasionally be reported as slightly greater than 100% because of inexact sampling times and differences in the accuracy between the system clock and the modem clock. If the percentage of use greatly exceeds 100%, or never exceeds 50%, then the baud rate set for the device probably does not reflect the speed of the modem.

NAME	sys-unconfig – undo a system’s configuration
SYNOPSIS	/usr/sbin/sys-unconfig
AVAILABILITY	SUNWadmap
DESCRIPTION	<p>sys-unconfig packs up a machine to make it ready to be configured again. It restores a system’s configuration to an “as-manufactured” state. A system’s configuration consists of hostname, Network Information Service (NIS) domain name, timezone, IP address, IP subnet mask, and root password. This operation is the inverse of those performed by the sysidnet, sysidnis, and sysidsys programs run at boot. (See sysidtool(1M)).</p> <p>sys-unconfig does the following:</p> <ul style="list-style-type: none"> • Saves current /etc/inet/hosts file information in /etc/inet/hosts.saved. • Restores the default /etc/inet/hosts file. • Removes the default hostname in /etc/hostname.?[0-9] and /etc/nodename. • Removes the default domainname in /etc/defaultdomain. • Restores the timezone to PST8PDT in /etc/TIMEZONE. • Disables the Network Information Service (NIS) and Network Information Service Plus (NIS+) if either NIS or NIS+ was configured. • Removes the entries for this host in /etc/net/*/hosts. • Removes the file /etc/inet/netmasks. • Removes the password set for root in /etc/shadow. • Execute all system configuration applications. These applications are defined by prior executions of a sysidconfig –a application. (See sysidconfig(1M)). When sys-unconfig is run, all system configuration applications are passed one argument, -u. <p>When sys-unconfig is finished, it performs a system shutdown.</p> <p>sys-unconfig is a potentially dangerous utility and can only be run by the super-user.</p>
FILES	/etc/default/init /etc/defaultdomain /etc/hostname.?[0-9] /etc/inet/hosts /etc/inet/netmasks /etc/net/*/hosts /etc/nodename /etc/shadow /var/nis/NIS_COLD_START /var/yp/binding/*/ypservers

SEE ALSO

sysidconfig(1M), **sysidtool(1M)**

BUGS

sys-unconfig is not available on diskless or dataless systems.

NAME	sysdef – output system definition
SYNOPSIS	<code>/usr/sbin/sysdef [-n <i>namelist</i>]</code> <code>/usr/sbin/sysdef [-h] [-d]</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>sysdef outputs the current system definition in tabular form. It lists all hardware devices, as well as pseudo devices, system devices, loadable modules, and the values of selected kernel tunable parameters.</p> <p>It generates the output by analyzing the named bootable operating system file (<i>namelist</i>) and extracting the configuration information from it.</p> <p>The default system <i>namelist</i> is <code>/dev/kmem</code>.</p>
OPTIONS	<p><code>-n <i>namelist</i></code> Specifies a <i>namelist</i> other than the default (<code>/dev/kmem</code>). The <i>namelist</i> specified must be a valid bootable operating system.</p> <p><code>-h</code> Prints the identifier of the current host in hexadecimal. This numeric value is unique across all Sun hosts.</p> <p><code>-d</code> The output includes the configuration of system peripherals formatted as a device tree.</p>
EXAMPLES	<p>The following example displays the format of the <code>sysdef -d</code> output:</p> <pre> example% sysdef -d Node 'Sun 4/60', unit #0 (no driver) Node 'options', unit #0 (no driver) Node 'zs', unit #0 Node 'zs', unit #1 Node 'fd', unit #0 Node 'audio', unit #0 Node 'sbus', unit #0 Node 'dma', unit #0 Node 'esp', unit #0 Node 'st', unit #1 (no driver) Node 'st', unit #0 Node 'sd', unit #2 Node 'sd', unit #1 Node 'sd', unit #0 Node 'le', unit #0 Node 'bwtwo', unit #0 Node 'auxiliary-io', unit #0 Node 'interrupt-enable', unit #0 Node 'memory-error', unit #0 Node 'counter-timer', unit #0 Node 'eeprom', unit #0 </pre>

FILES /dev/kmem default operating system image

SEE ALSO **hostid(1)**, **prtconf(1M)**, **nlist(3E)**

NAME	sysidconfig – execute system configuration applications, or define set of system configuration applications
SYNOPSIS	sysidconfig [-b <i>basedir</i>] [-l] [-a <i>application</i>] [-r <i>application</i>] [-v]
AVAILABILITY	SUNWadmap
DESCRIPTION	<p>Invoked without any options, the sysidconfig program executes a list of applications. An application on this list is referred to as a "system configuration application." Every application on this list will be passed one command-line argument, -c. This flag will cause the system configuration application to perform its configuration function. Without options, sysidconfig should only be invoked by startup scripts, which occurs during the initial installation and during a reconfigure reboot.</p> <p>All applications on the list will be executed, if possible. All activity taken by the sysidconfig program is logged in the sysidconfig log file, /var/log/sysidconfig.log. If one or more of the applications on the list are either not present at execution time, are not executable, or execute but return a failure code upon completion, then that information will be logged as well. Successful completion of the program can be assumed if no error message is present in the log file. Programs are executed sequentially, with only one configuration application active at a time.</p> <p>Executed with the -l, -a, or -r options, the sysidconfig program allows the superuser to list the defined configuration applications, and to add items to or remove items from that list. Running sysidconfig with options is the only way to view or manipulate the list. Only the superuser can execute the sysidconfig program with options.</p> <p>The -b and -v options change the behavior of sysidconfig, and can be used with or without the list manipulation options discussed above. The -b <i>basedir</i> option is used to specify a reference root directory other than the default, /. The -v option duplicates the log file output on stdout.</p> <p>By default, no SPARC applications exist on this list. However, the x86 system is delivered with one application, kdmconfig(1M), on the list. kdmconfig is an x86-specific program that is not delivered on SPARC systems.</p> <p>This application is an extension of the sysidtool(1M) suite of programs. It is executed during initial installation and during a reconfigure reboot, before the window system has been started. Graphical User Interface (GUI) applications will not execute successfully if they are added to the list of configuration applications via sysidconfig -a.</p> <p>This program is referenced, but not fully described, in the sysidtool(1M) manual page.</p>
OPTIONS	<p>The valid options are:</p> <p>-b <i>basedir</i> Specify an alternate base directory (/ is defined as the default base directory if no other is specified). The specified directory is used as the root directory when adding, listing, removing, or executing configuration applications. The log file where information is recorded is in /var/log, relative to the specified <i>basedir</i>. In the log file, the <i>basedir</i> is not noted.</p>

This means, for example, that if the superuser on a diskless client's server executes:

```
sysidconfig -b /export/root/client -a /sbin/someapp
```

then the diskless client *client* would have */sbin/someapp* executed upon reconfigure reboot. The diskless client's log file would note that */sbin/someapp* was added, not */export/root/client/sbin/someapp*.

- l** List defined configuration applications. Applications will be executed one at a time, in the order shown in the list.
- a application** Add the named application to the list of defined applications. When next invoked without arguments, **sysidconfig** will run this newly added application after all previously defined applications. *application* must be a fully qualified path name that is not currently on the list of applications to execute.
- r application** Remove the named application from the list of defined applications. *application* must be a fully qualified path name and it must be on the existing list of applications to execute.
- v** Verbose mode. This option echoes all information sent to the log file to **stdout**. Such information includes timestamp information about when the program was executed, the names of applications being executed, and results of those executions.

RETURN VALUES

The **sysidconfig** program will return **0** if it completes successfully.

When executed with the **-r** or **-a** options, error conditions or warnings will be reported on **stderr**. If the requested action completes successfully, an exit code of **0** will be returned.

ERRORS

- EPERM** The program was executed by a user other than the superuser.
- EINVAL** Option **-l**, **-a**, or **-r** was passed and the action could not be completed successfully.

FILES

/var/log/sysidconfig.log

SEE ALSO

sys-unconfig(1M), **sysidtool(1M)**

x86 Only

kdmconfig(1M)

DIAGNOSTICS

When run without options, a log of the **sysidconfig** program's activity can be found in **/var/log/sysidconfig.log**. This file contains a timestamp log of each program executed, its resulting **stderr** output, and its exit code. If an application in the list was not found or is not executable, that will also be noted.

NAME	sysidtool, sysidnet, sysidnis, sysidsys, sysidroot – system configuration
SYNOPSIS	<pre> /usr/sbin/sysidnet /usr/sbin/sysidnis /usr/sbin/sysidsys /usr/sbin/sysidroot </pre>
AVAILABILITY	SUNWadmap
DESCRIPTION	<p>sysidtool is a suite of four programs that configure a new system, or one that has been unconfigured with sys-unconfig(1M). The sysidtool programs run automatically at system installation, or during the first boot after a machine has been successfully unconfigured.</p> <p>These programs have no effect except at such times, and should never be run manually. The sysidtool programs set up the appropriate information in the machine's configuration files, in the kernel, and on the machine's network interface. They may prompt for the following information:</p> <p>sysidnet: network configuration</p> <ul style="list-style-type: none"> • Machine's default locale • Machine's console type • Machine's host name • Machine's IP address <p>sysidnis: NIS/NIS+ client configuration</p> <ul style="list-style-type: none"> • Name service choice: NIS+, NIS, or none • Machine's IP subnet mask (if no NIS/NIS+ server can automatically be located on the machine's sub-network) • NIS/NIS+ domain name • Hostname and IP address of an NIS/NIS+ server in the domain <p>sysidsys: miscellaneous system configuration</p> <ul style="list-style-type: none"> • Machine's IP subnet mask (if an NIS/NIS+ server was automatically located on the machine's sub-network) • Machine's time zone • Date and time <p>sysidroot: control superuser information</p> <ul style="list-style-type: none"> • Machine's root password <p>sysidconfig: host or platform-specific configuration</p> <ul style="list-style-type: none"> • This command controls specification and execution of custom configuration applications which may be specified for a particular host or a particular platform. (See sysidconfig(1M)). <p>The sysidtool programs attempt to obtain configuration information from various network services and databases, as set up by the Host Manager function of admintool(1M). Only when information is missing from these services and databases, do the sysidtool programs prompt the user for the information. However, the user is always prompted</p>

for the system's root password.

The machine's configuration information is set up in its **/etc** and **/var** files.

FILES

/etc/.UNCONFIGURED

/etc/nodename

/etc/hostname.?[0-9]

/etc/default/init

/etc/defaultdomain

/etc/passwd

password file. See **passwd(4)**.

/etc/shadow

shadow password file. See **shadow(4)**.

/etc/inet/hosts

/etc/inet/netmasks

/etc/net*/hosts

/var/nis/NIS_COLD_START

/var/yp/aliases

/var/yp/binding*/ypservers

/etc/.sysIDtool.state

SEE ALSO

admintool(1M), **sysidconfig(1M)**, **sys-unconfig(1M)**, **passwd(4)**, **shadow(4)**

NOTES

If a system has more than one network interface, you can use **sysidtool** to configure only the primary interface on the system. All other interfaces on the system must be configured manually.

NAME	syslogd – log system messages
SYNOPSIS	<code>/usr/sbin/syslogd [-d] [-f <i>configfile</i>] [-m <i>markinterval</i>] [-p <i>path</i>]</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>syslogd reads and forwards system messages to the appropriate log files and/or users, depending upon the priority of a message and the system facility from which it originates. The configuration file <code>/etc/syslog.conf</code> (see syslog.conf(4)) controls where messages are forwarded. syslogd logs a mark (timestamp) message every <i>markinterval</i> minutes (default 20) at priority <code>LOG_INFO</code> to the facility whose name is given as mark in the <code>syslog.conf</code> file.</p> <p>A system message consists of a single line of text, which may be prefixed with a priority code number enclosed in angle-brackets (< >); priorities are defined in <code><sys/syslog.h></code>.</p> <p>syslogd reads from the STREAMS log driver, <code>/dev/log</code>, and from any transport provider specified in <code>/etc/netconfig</code>, <code>/etc/net/transport/hosts</code>, and <code>/etc/net/transport/services</code>.</p> <p>syslogd reads the configuration file when it starts up, and again whenever it receives a HUP signal (see signal(5), at which time it also closes all files it has open, re-reads its configuration file, and then opens only the log files that are listed in that file. syslogd exits when it receives a TERM signal.</p> <p>As it starts up, syslogd creates the file <code>/etc/syslog.pid</code>, if possible, containing its process identifier (PID).</p>
OPTIONS	<p><code>-d</code> Turn on debugging.</p> <p><code>-f <i>configfile</i></code> Specify an alternate configuration file.</p> <p><code>-m <i>markinterval</i></code> Specify an interval, in minutes, between mark messages.</p> <p><code>-p <i>path</i></code> Specify an alternative log device name. The default is <code>/dev/log</code>.</p>
FILES	<p><code>/etc/syslog.conf</code> configuration file</p> <p><code>/etc/syslog.pid</code> process ID</p> <p><code>/dev/log</code> STREAMS log driver</p> <p><code>/etc/netconfig</code> specifies the transport providers available on the system</p> <p><code>/etc/net/transport/hosts</code> network hosts for each transport</p> <p><code>/etc/net/transport/services</code> network services for each transport</p>
SEE ALSO	logger(1) , syslog(3) , syslog.conf(4) , signal(5) , log(7D)

NAME	tapes – adds /dev entries for tape drives attached to the system
SYNOPSIS	<code>/usr/sbin/tapes [-r rootdir]</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>tapes creates symbolic links from the <code>/dev/rmt</code> directory to the actual tape device special files under the <code>/devices</code> directory tree. tapes searches the kernel device tree to see what tape devices are attached to the system. For each equipped tape drive, the following steps are performed:</p> <ol style="list-style-type: none"> 1. The <code>/dev/rmt</code> directory is searched for a <code>/dev/rmt/n</code> entry that is a symbolic link to the <code>/device</code> special node of the current tape drive. If one is found, this determines the logical controller number of the tape drive. 2. The rest of the special devices associated with the drive are then checked, and incorrect symbolic links are removed and necessary ones added. 3. If none are found, a new logical controller number is assigned (the lowest-unused number), and new symbolic links are created for all the special devices associated with the drive. <p>tapes does not remove unneeded tape drive links; these must be removed by hand.</p> <p>tapes is called each time the system is reconfiguration-booted. tapes can only be run after drvconfig(1M) is run, since drvconfig builds the kernel data structures and the <code>/devices</code> tree.</p> <p>Of interest to device driver writers are the devices that tapes considers tape devices. Device nodes of type <code>DDI_NT_TAPE</code> are considered tape devices (see ddi_create_minor_node(9F) for more information on node types.)</p>
OPTIONS	<code>-r rootdir</code> tapes presumes that the <code>/dev/rmt</code> and <code>/devices</code> directories are found under <code>rootdir</code> , not directly under <code>/</code> .
ERRORS	If tapes finds entries of a particular logical controller linked to different physical controllers, it prints an error message and exits without making any changes to the <code>/dev</code> directory, since it cannot determine which of the two alternative logical to physical mappings is correct. The links should be manually corrected or removed before another reconfiguration boot is performed.
FILES	<code>/dev/rmt/*</code> entries for the tape drive for general use <code>/devices/*</code> device nodes
SEE ALSO	devlinks (1M), disks (1M), drvconfig (1M), ports (1M), ddi_create_minor_node (9F)

NAME	taskstat – prints ASET tasks status
SYNOPSIS	<code>/usr/aset/util/taskstat [-d <i>aset_dir</i>]</code>
DESCRIPTION	<p>taskstat is located in the <code>/usr/aset/util</code> directory. <code>/usr/aset</code> is the default operating directory of the Automated Security Enhancement Tool (ASET). An alternative working directory can be specified by the administrators through the aset -d command or the ASETDIR environment variable. See aset(1M). Because aset dispatches its tasks to run in the background, when it returns, these tasks may or may not have completed. taskstat prints the status of the tasks, listing those that are completed and those that are still executing.</p> <p>The ASET reports, which are located in the <code>/usr/aset/reports</code> directory (see the -d option), are not complete until all the tasks finish executing.</p>
OPTIONS	<p>-d <i>aset_dir</i> Specify the working directory for ASET. By default, this directory is <code>/usr/aset</code>. With this option, the reports directory will be located under <i>aset_dir</i>.</p>
SEE ALSO	<p>aset(1M) <i>System Administration Guide, Volume I</i></p>

NAME	tcxconfig – configure the default linearity of the 24-bit TrueColor Visual for OpenWindows on a system with an S24 frame buffer (TCX)				
SYNOPSIS	<code>/usr/sbin/tcxconfig [linear nonlinear]</code>				
AVAILABILITY	SUNWtcxow				
DESCRIPTION	<p>The tcxconfig script changes the default linearity of a 24-bit TrueColor Visual for OpenWindows on a system with an S24 frame buffer. When the S24 graphics driver for OpenWindows is installed, the default 24-bit TrueColor Visual is nonlinear. You can run tcxconfig with an argument that specifies the setting you want.</p> <p>OpenWindows should not be running when you execute the tcxconfig script with an option. Start OpenWindows after tcxconfig has set the linearity you desire.</p>				
OPTIONS	<p>If you specify no option, tcxconfig displays the current default setting. You must become superuser before you can execute tcxconfig with one of the following options.</p> <table><tr><td>linear</td><td>Set linear visual to be the default 24-bit TrueColor Visual. This means colors will be gamma-corrected.</td></tr><tr><td>nonlinear</td><td>Set nonlinear visual to be the default 24-bit TrueColor Visual.</td></tr></table>	linear	Set linear visual to be the default 24-bit TrueColor Visual. This means colors will be gamma-corrected.	nonlinear	Set nonlinear visual to be the default 24-bit TrueColor Visual.
linear	Set linear visual to be the default 24-bit TrueColor Visual. This means colors will be gamma-corrected.				
nonlinear	Set nonlinear visual to be the default 24-bit TrueColor Visual.				
EXIT STATUS	<p>The following exit values are returned:</p> <table><tr><td>0</td><td>success</td></tr><tr><td>1</td><td>an error has occurred.</td></tr></table>	0	success	1	an error has occurred.
0	success				
1	an error has occurred.				

NAME	tic – terminfo compiler
SYNOPSIS	tic [-v[n]] [-c] <i>file</i>
DESCRIPTION	<p>The command tic translates a terminfo file from the source format into the compiled format. The results are placed in the directory /usr/share/lib/terminfo. The compiled format is necessary for use with the library routines in curses(3X).</p> <p>If the environment variable TERMINFO is set, the compiled results are placed there instead of /usr/share/lib/terminfo.</p> <p>Total compiled entries cannot exceed 4096 bytes. The name field cannot exceed 128 bytes. Terminal names exceeding 14 characters will be truncated to 14 characters and a warning message will be printed.</p>
OPTIONS	<p>-vn Specify that (verbose) output be written to standard error trace information showing tic's progress. The optional integer <i>n</i> is a number from 1 to 10, indicating the desired level of detail of information. If <i>n</i> is omitted, the default level is 1. If <i>n</i> is specified and greater than 1, the level of detail is increased.</p> <p>-c Specifies to check only <i>file</i> for errors. Errors in use= links are not detected.</p> <p><i>file</i> Contains one or more terminfo terminal descriptions in source format [see terminfo(4)]. Each description in the file describes the capabilities of a particular terminal. When a use=entry-name field is discovered in a terminal entry currently being compiled, tic reads in the binary from /usr/share/lib/terminfo to complete the entry. (Entries created from <i>file</i> will be used first. If the environment variable TERMINFO is set, that directory is searched instead of /usr/share/lib/terminfo.) tic duplicates the capabilities in <i>entry-name</i> for the current entry, with the exception of those capabilities that are explicitly defined in the current entry.</p>
FILES	/usr/share/lib/terminfo/?/* compiled terminal description database
SEE ALSO	captainfo(1M) , infocmp(1M) , curses(3X) , terminfo(4)
NOTES	When an entry, for example, entry_name_1 , contains a use=entry_name_2 field, any canceled capabilities in <i>entry_name_2</i> must also appear in entry_name_1 before use= for these capabilities to be canceled in entry_name_1 .

NAME	ttyadm – format and output port monitor-specific information
SYNOPSIS	<pre> /usr/sbin/ttyadm [-b] [-c] [-h] [-I] [-r count] [-i msg] [-m modules] [-p prompt] [-t timeout] [-S y n] [-T termtype] -d device -l ttylabel -s service /usr/sbin/ttyadm -V </pre>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>The ttyadm command is an administrative command that formats ttymon(1M)-SPECIFIC information and writes it to standard output. The Service Access Facility (SAF) requires each port monitor to provide such a command. Note that the port monitor administrative file is updated by the Service Access Controller's administrative commands, sacadm(1M) and pmadm(1M). ttyadm provides a means of presenting formatted port monitor-specific (ttymon-specific) data to these commands.</p>
OPTIONS	<p>-b Set the “bi-directional port” flag. When this flag is set, the line can be used in both directions. ttymon will allow users to connect to the service associated with the port, but if the port is free, uucico(1M), cu(1C), or ct(1C) can use it for dialing out.</p> <p>-c Set the connect-on-carrier flag for the port. If the -c flag is set, ttymon will invoke the port's associated service immediately when a connect indication is received (that is, no prompt is printed and no baud-rate searching is done).</p> <p>-h Set the hangup flag for the port. If the -h flag is not set, ttymon will force a hangup on the line by setting the speed to 0 before setting the speed to the default or specified value.</p> <p>-I Initialize only. If the -I option is used, ttymon will invoke the service only once. This can be used to configure a particular device without actually monitoring it, as with software carrier.</p> <p>-d device <i>device</i> is the full pathname of the device file for the TTY port.</p> <p>-i message Specify the inactive (disabled) response message. This message will be sent to the TTY port if the port is disabled or the ttymon monitoring the port is disabled.</p> <p>-l ttylabel Specify which <i>ttylabel</i> in the /etc/ttydefs file to use as the starting point when searching for the proper baud rate.</p> <p>-m modules Specify a list of pushable STREAMS modules. The modules will be pushed in the order in which they are specified before the service is invoked. <i>modules</i> must be a comma-separated list of modules, with no white space included. Any modules currently on the stream will be popped before these modules are pushed.</p> <p>-p prompt Specify the prompt message, for example, “login:”.</p>

- r** *count* When the **-r** option is invoked, **ttymon** will wait until it receives data from the port before it displays a prompt. If *count* is **0**, **ttymon** will wait until it receives any character. If *count* is greater than **0**, **ttymon** will wait until *count* newlines have been received.
- s** *service* *service* is the full pathname of the service to be invoked when a connection request is received. If arguments are required, the command and its arguments must be enclosed in double quotes (" ").
- t** *timeout* Specify that **ttymon** should close a port if the open on the port succeeds, and no input data is received in *timeout* seconds.
- S y | n** Set the software carrier value. **y** will turn software carrier on. **n** will turn software carrier off.
- T** *termttype* Set the terminal type. The **TERM** environment variable will be set to *termttype*.
- V** Display the version number of the current **/usr/lib/saf/ttymon** command.

OUTPUT If successful, **tyadm** will generate the requested information, write it to standard output, and exit with a status of **0**. If **tyadm** is invoked with an invalid number of arguments or invalid arguments, or if an incomplete option is specified, an error message will be written to standard error and **ttymon** will exit with a non-zero status.

FILES */etc/ttydefs*

SEE ALSO *ct(1C)*, *cu(1C)*, *pmadm(1M)*, *sacadm(1M)*, *ttymon(1M)*, *uucico(1M)*
System Administration Guide, Volume II

NAME	ttyhstmgr – administer hosts on a network
SYNOPSIS	/usr/bin/ttyhstmgr
DESCRIPTION	<p>ttyhstmgr offers a curses-based interface that supports the definition of client machines on the server machine. Membership in the sysadmin group (gid 14) is used to restrict access to this tool. Members of the sysadmin group can add a client to a server, or delete a client from a server. Non-members have read-only permissions, and can use ttyhstmgr to get information about a host on the network.</p> <p>Three types of clients are supported:</p> <ul style="list-style-type: none">diskless Clients that depend on the server for all disk needs.standalone Clients that may use the server as an optional file server.dataless Clients that have a local disk but use it only for root and swap. <p>The supported naming services are:</p> <ul style="list-style-type: none">NIS+ Network Information Service Plus, the replacement for NIS.NIS Network Information Service, formerly known as YP.None Text files in the /etc directory.
SEE ALSO	nis+(1) , admintool(1M) <i>SPARC: Installing Solaris Software</i> <i>x86: Installing Solaris Software</i>

NAME	ttymon – port monitor for terminal ports
SYNOPSIS	<pre> /usr/lib/saf/ttymon /usr/lib/saf/ttymon -g [-d device] [-h] [-t timeout] [-l ttylabel] [-p prompt] [-m modules] [-T termtype] </pre>
DESCRIPTION	<p>ttymon is a STREAMS-based TTY port monitor. Its function is to monitor ports, to set terminal modes, baud rates, and line disciplines for the ports, and to connect users or applications to services associated with the ports. Normally, ttymon is configured to run under the Service Access Controller, sac(1M), as part of the Service Access Facility (SAF). It is configured using the sacadm(1M) command. Each instance of ttymon can monitor multiple ports. The ports monitored by an instance of ttymon are specified in the port monitor's administrative file. The administrative file is configured using the pmadm(1M) and tyadm(1M) commands. When an instance of ttymon is invoked by the sac command, it starts to monitor its ports. For each port, ttymon first initializes the line disciplines, if they are specified, and the speed and terminal settings. For ports with entries in /etc/logindevperm, device owner, group and permissions are set. (See logindevperm(4).) The values used for initialization are taken from the appropriate entry in the TTY settings file. This file is maintained by the sttydefs(1M) command. Default line disciplines on ports are usually set up by the autopush(1M) command of the Autopush Facility.</p> <p>ttymon then writes the prompt and waits for user input. If the user indicates that the speed is inappropriate by pressing the BREAK key, ttymon tries the next speed and writes the prompt again. When valid input is received, ttymon interprets the per-service configuration file for the port, if one exists, creates a utmp entry if required (see utmp(4)), establishes the service environment, and then invokes the service associated with the port. Valid input consists of a string of at least one non-newline character, terminated by a carriage return. After the service terminates, ttymon cleans up the utmp entry, if one exists, and returns the port to its initial state.</p> <p>If autobaud is enabled for a port, ttymon will try to determine the baud rate on the port automatically. Users must enter a carriage return before ttymon can recognize the baud rate and print the prompt. Currently, the baud rates that can be determined by autobaud are 110, 1200, 2400, 4800, and 9600.</p> <p>If a port is configured as a bidirectional port, ttymon will allow users to connect to a service, and, if the port is free, will allow uucico(1M), cu(1C), or ct(1C) to use it for dialing out. If a port is bidirectional, ttymon will wait to read a character before it prints a prompt.</p> <p>If the <i>connect-on-carrier</i> flag is set for a port, ttymon will immediately invoke the port's associated service when a connection request is received. The prompt message will not be sent.</p>

If a port is disabled, **tytmon** will not start any service on that port. If a disabled message is specified, **tytmon** will send out the disabled message when a connection request is received. If **tytmon** is disabled, all ports under that instance of **tytmon** will also be disabled.

SERVICE INVOCATION

The service **tytmon** invokes for a port is specified in the **tytmon** administrative file. **tytmon** will scan the character string giving the service to be invoked for this port, looking for a **%d** or a **%%** two-character sequence. If **%d** is found, **tytmon** will modify the service command to be executed by replacing those two characters by the full path name of this port (the device name). If **%%** is found, they will be replaced by a single **%**.

When the service is invoked, file descriptor **0**, **1**, and **2** are opened to the port device for reading and writing. The service is invoked with the user ID, group ID and current home directory set to that of the user name under which the service was registered with **tytmon**. Two environment variables, **HOME** and **TTYPROMPT**, are added to the service's environment by **tytmon**. **HOME** is set to the home directory of the user name under which the service is invoked. **TTYPROMPT** is set to the prompt string configured for the service on the port. This is provided so that a service invoked by **tytmon** has a means of determining if a prompt was actually issued by **tytmon** and, if so, what that prompt actually was.

See **tytadm(1M)** for options that can be set for ports monitored by **tytmon** under the Service Access Controller.

OPTIONS

- g** A special invocation of **tytmon** is provided with the **-g** option. This form of the command should only be called by applications that need to set the correct baud rate and terminal settings on a port and then connect to **login** service, but that cannot be pre-configured under the SAC. The following combinations of options can be used with **-g**:
- d device** *device* is the full path name of the port to which **tytmon** is to attach. If this option is not specified, file descriptor **0** must be set up by the invoking process to a TTY port.
- h** If the **-h** flag is not set, **tytmon** will force a hangup on the line by setting the speed to zero before setting the speed to the default or specified speed.
- l ttylabel** *ttylabel* is a link to a speed and TTY definition in the **tytdefs** file. This definition tells **tytmon** at what speed to run initially, what the initial TTY settings are, and what speed to try next if the user indicates that the speed is inappropriate by pressing the BREAK key. The default speed is 9600 baud.
- m modules** When initializing the port, **tytmon** will pop all modules on the port, and then push *modules* in the order specified. *modules* is a comma-separated list of pushable modules. Default modules on the ports are usually set up by the Autopush Facility.
- p prompt** Allows the user to specify a prompt string. The default prompt is **Login**:

- t *timeout*** Specifies that **ttypmon** should exit if no one types anything in *timeout* seconds after the prompt is sent.
-T *termtype* Sets the **TERM** environment variable to *termtype*.

ENVIRONMENT

If any of the **LC_*** variables (**LC_CTYPE**, **LC_MESSAGES**, **LC_TIME**, **LC_COLLATE**, **LC_NUMERIC**, and **LC_MONETARY**) (see **environ(5)**) are not set in the environment, the operational behavior of **ttypmon** for each corresponding locale category is determined by the value of the **LANG** environment variable. If **LC_ALL** is set, its contents are used to override both the **LANG** and the other **LC_*** variables. If none of the above variables is set in the environment, the "C" (U.S. style) locale determines how **ttypmon** behaves.

LC_CTYPE

Determines how **ttypmon** handles characters. When **LC_CTYPE** is set to a valid value, **ttypmon** can display and handle text and filenames containing valid characters for that locale. **ttypmon** can display and handle Extended Unix Code (EUC) characters where any individual character can be 1, 2, or 3 bytes wide. **ttypmon** can also handle EUC characters of 1, 2, or more column widths. In the "C" locale, only characters from ISO 8859-1 are valid.

FILES

/etc/logindevperm

SEE ALSO

ct(1C), **cu(1C)**, **autopush(1M)**, **pmadm(1M)**, **sac(1M)**, **sacadm(1M)**, **sttydefs(1M)**, **ttypadm(1M)**, **uucico(1M)**, **logindevperm(4)**, **utmp(4)**, **environ(5)**

System Administration Guide, Volume II

NOTES

If a port is monitored by more than one **ttypmon**, it is possible for the **ttypmons** to send out prompt messages in such a way that they compete for input.

NAME	tunefs – tune up an existing file system
SYNOPSIS	tunefs [-a <i>maxcontig</i>] [-d <i>rotdelay</i>] [-e <i>maxbpg</i>] [-m <i>minfree</i>] [-o [space time]] <i>special</i> <i>filesystem</i>
DESCRIPTION	<p>tunefs is designed to change the dynamic parameters of a file system which affect the layout policies. The file system must be unmounted before using tunefs. When using tunefs with filesystem, filesystem must be in /etc/vfstab. The parameters which are to be changed are indicated by the options given below.</p> <p>Generally one should optimize for time unless the file system is over 90% full.</p>
OPTIONS	<p>-a <i>maxcontig</i> Specify the maximum number of contiguous blocks that will be laid out before forcing a rotational delay (see -d below). The default value is 1, since most device drivers require an interrupt per disk transfer. Device drivers that can chain several buffers together in a single transfer should set this to the maximum chain length.</p> <p>-d <i>rotdelay</i> Specify the expected time (in milliseconds) to service a transfer completion interrupt and initiate a new transfer on the same disk. It is used to decide how much rotational spacing to place between successive blocks in a file.</p> <p>-e <i>maxbpg</i> Indicate the maximum number of blocks any single file can allocate out of a cylinder group before it is forced to begin allocating blocks from another cylinder group. Typically this value is set to approximately one quarter of the total blocks in a cylinder group. The intent is to prevent any single file from using up all the blocks in a single cylinder group, thus degrading access times for all files subsequently allocated in that cylinder group. The effect of this limit is to cause big files to do long seeks more frequently than if they were allowed to allocate all the blocks in a cylinder group before seeking elsewhere. For file systems with exclusively large files, this parameter should be set higher.</p> <p>-m <i>minfree</i> Specify the percentage of space held back from normal users; the minimum free space threshold. The default value used is 10%. This value can be set to 0, however up to a factor of three in throughput will be lost over the performance obtained at a 10% threshold. Note: If the value is raised above the current usage level, users will be unable to allocate files until enough files have been deleted to get under the higher threshold.</p> <p>-o [space time] Change optimization strategy for the file system.</p> <p>space: conserve space time: attempt to organize file layout to minimize access time</p>

SEE ALSO | **mkfs(1M), fork(2), terminfo(4)**

NAME	uadmin – administrative control
SYNOPSIS	<i>/sbin/uadmin cmd fcn</i>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>The uadmin command provides control for basic administrative functions. This command is tightly coupled to the System Administration procedures and is not intended for general use. It may be invoked only by the super-user.</p> <p>Both <i>cmd</i> (command) and <i>fcn</i> (function) are converted to integers and passed to the uadmin system call.</p>
SEE ALSO	uadmin(2)

NAME	ufsdump – incremental file system dump
SYNOPSIS	<i>/usr/sbin/ufsdump</i> [<i>options</i>] [<i>arguments</i>] <i>files_to_dump</i>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>ufsdump backs up all files specified by <i>files_to_dump</i> (normally either a whole file system or files within a file system changed after a certain date) to magnetic tape, diskette, or disk file. When running ufsdump, the file system must be inactive; otherwise, the output of ufsdump may be inconsistent and restoring files correctly may be impossible. A file system is inactive when it is unmounted or the system is in single user mode. A file system is not considered inactive if one tree of the file system is quiescent while another tree has files or directories being modified.</p> <p><i>options</i> is a single string of one-letter ufsdump options.</p> <p><i>arguments</i> may be multiple strings whose association with the options is determined by order. That is, the first argument goes with the first option that takes an argument; the second argument goes with the second option that takes an argument, and so on.</p> <p><i>files_to_dump</i> is required and must be the last argument on the command line. See OPERANDS for more information.</p> <p>With most devices ufsdump can automatically detect the end-of-media. Consequently, the d, s, and t options are not necessary for multi-volume dumps, unless ufsdump does not understand the way the device detects the end-of-media, or the files are to be restored on a system with an older version of the restore command.</p>
OPTIONS	<p>The following options are supported:</p> <p>0–9 The “dump level.” All files specified by <i>files_to_dump</i> that have been modified since the last ufsdump at a lower dump level are copied to the <i>dump_file</i> destination (normally a magnetic tape device). For instance, if a “level 2” dump was done on Monday, followed by a “level 4” dump on Tuesday, a subsequent “level 3” dump on Wednesday would contain all files modified or added since the “level 2” (Monday) backup. A “level 0” dump copies the entire file system to the <i>dump_file</i>.</p> <p>a <i>archive_file</i> Archive file. Archive a dump table-of-contents in the specified <i>archive_file</i> to be used by ufsrestore(1M) to determine whether a file is in the dump file that is being restored.</p> <p>b <i>factor</i> Blocking factor. Specify the blocking factor for tape writes. The default is 20 blocks per write for tapes of density less than 6250BPI (bytes-per-inch). The default blocking factor for tapes of density 6250BPI and greater is 64. The default blocking factor for cartridge tapes (c option) is 126. The highest blocking factor available with most tape drives is 126. Note: the blocking factor is specified in terms of 512-byte blocks, for compatibility with tar(1).</p>

- c** Cartridge. Set the defaults for cartridge instead of the standard half-inch reel. This sets the density to 1000BPI and the blocking factor to 126. Since **ufsdump** can automatically detect the end-of-media, only the blocking parameter normally has an effect. When cartridge tapes are used, and this option is *not* specified, **ufsdump** will slightly miscalculate the size of the tape. If the **b**, **d**, **s** or **t** options are specified with this option, their values will override the defaults set by this option.
- d** *bpi* Tape density. Not normally required, as **ufsdump** can detect end-of-media. This parameter can be used to keep a running tab on the amount of tape used per reel. The default density is 6250BPI except when the **c** option is used for cartridge tape, in which case it is assumed to be 1000BPI per track. Typical values for tape devices are:

1/2" tape	6250 BPI
1/4" cartridge	1000 BPI

The tape densities and other options are documented in the **st(7D)** man page.
- D** Diskette. Dump to diskette.
- f** *dump_file* Dump file. Use *dump_file* as the file to dump to, instead of **/dev/rmt/0**. If *dump_file* is specified as **-**, dump to standard output.

If the name of the file is of the form *machine:device*, the dump is done from the specified machine over the network using **rmt(1M)**. Since **ufsdump** is normally run by root, the name of the local machine must appear in the **.rhosts** file of the remote machine. If the file is specified as *user@machine:device*, **ufsdump** will attempt to execute as the specified user on the remote machine. The specified user must have a **.rhosts** file on the remote machine that allows the user invoking the command from the local machine to access the remote machine.
- l** Autoload. When the end-of-tape is reached before the dump is complete, take the drive offline 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, prompt for another tape and wait.
- n** Notify all operators in the **sys** group that **ufsdump** requires attention by sending messages to their terminals, in a manner similar to that used by the **wall(1M)** command. Otherwise, such messages are sent only to the terminals (such as the console) on which the user running **ufsdump** is logged in.
- o** Offline. Take the drive offline when the dump is complete or the end-of-media is reached and rewind the tape, or eject the diskette. In the case of some autoloading 8mm drives, the tape is removed from the drive automatically. This prevents another process which rushes in to use the drive, from inadvertently overwriting the media.

- s size** Specify the *size* of the volume being dumped to. Not normally required, as **ufsdump** can detect end-of-media. When the specified size is reached, **ufsdump** waits for you to change the volume. **ufsdump** interprets the specified size as the length in feet for tapes and cartridges, and as the number of 1024-byte blocks for diskettes. The values should be a little smaller than the actual physical size of the media (for example, 425 for a 450-foot cartridge). Typical values for tape devices depend on the **c** option, for cartridge devices, and the **D** option for diskettes:
- | | |
|--------------------------|---|
| 1/2" tape | 2300 feet |
| 60-Mbyte 1/4" cartridge | 425 feet |
| 150-Mbyte 1/4" cartridge | 700 feet |
| diskette | 1422 blocks (Corresponds to a 1.44-Mbyte diskette, with one cylinder reserved for bad block information.) |
- S** Size estimate. Determine the amount of space that is needed to perform the dump without actually doing it, and display the estimated number of bytes it will take. This is useful with incremental dumps to determine how many volumes of media will be needed.
- t tracks** Specify the number of tracks for a cartridge tape. Not normally required, as **ufsdump** can detect end-of-media. The default is 9 tracks. The **t** option is not compatible with the **D** option. Values for Sun-supported tape devices are:
- | | |
|--------------------------|-----------|
| 60-Mbyte 1/4" cartridge | 9 tracks |
| 150-Mbyte 1/4" cartridge | 18 tracks |
- u** Update the dump record. Add an entry to the file **/etc/dumpdates**, for each file system successfully dumped that includes the file system name, date, and dump level.
- v** Verify. After each tape or diskette is written, verify the contents of the media against the source file system. If any discrepancies occur, prompt for new media, then repeat the dump/verification process. The file system *must* be unmounted. This option cannot be used to verify a dump to standard output.
- w** Warning. List the file systems that have not been backed up within a day. This information is gleaned from the files **/etc/dumpdates** and **/etc/vfstab**. When the **w** option is used, all other options are ignored. After reporting, **ufsdump** exits immediately.
- W** Warning with highlight. Similar to the **w** option, except that the **W** option includes all file systems that appear in **/etc/dumpdates**, along with information about their most recent dump dates and levels. File systems that have not been backed up within a day are highlighted.

OPERANDS

The following operand is supported:

files_to_dump Specifies the files to dump. Usually it identifies a whole file system by its raw device name (for example, **/dev/rdisk/c0t3d0s6**). Incremental dumps (levels **1** to **9**) of files changed after a certain date only apply to a whole file system. Alternatively, *files_to_dump* can identify individual files or directories. All files or directories are dumped, which is equivalent to a level **0** dump; however, **/etc/dumpdates** is not updated, even with the **u** option specified. In all cases, the files must be contained in the same file system, and the file system must be local to the system where **ufsdump** is being run.

files_to_dump is required and must be the last argument on the command line.

If no *options* are given, the default is **9uf /dev/rmt/0 files_to_dump**.

EXAMPLES

To make a full dump of a root file system on **c0t3d0**, on a 150-MByte cartridge tape unit **0**, use:

example# ufsdump 0cfu /dev/rmt/0 /dev/rdisk/c0t3d0s0

To make and verify an incremental dump at level **5** of the **usr** partition of **c0t3d0**, on a 1/2" reel tape unit **1**, use:

example# ufsdump 5fuv /dev/rmt/1 /dev/rdisk/c0t3d0s6

EXIT STATUS

While running, **ufsdump** emits many verbose messages. **ufsdump** returns the following exit values:

- 0** Normal exit.
- 1** Startup errors encountered.
- 3** Abort – no checkpoint attempted.

FILES

- /dev/rmt/0** default unit to dump to
- /etc/dumpdates** dump date record
- /etc/group** to find group **sys**
- /etc/hosts** to gain access to remote system with drive
- /etc/vfstab** list of file systems

SEE ALSO

cpio(1), **tar(1)**, **dd(1M)**, **devnm(1M)**, **prvtoc(1M)**, **rmt(1M)**, **shutdown(1M)**, **ufsrestore(1M)**, **volcopy(1M)**, **wall(1M)**, **wall(1M) st(7D)**

NOTES

Read Errors

Fewer than 32 read errors on the file system are ignored.

Process Per Reel

Because each reel requires a new process, parent processes for reels that are already written hang around until the entire tape is written.

**Operator
Intervention**

ufsdump requires operator intervention on these conditions: end of volume, end of dump, volume write error, volume open error or disk read error (if there are more than a threshold of 32). In addition to alerting all operators implied by the **n** option, **ufsdump** interacts with the operator on **ufsdump**'s control terminal at times when **ufsdump** can no longer proceed, or if something is grossly wrong. All questions **ufsdump** poses *must* be answered by typing **yes** or **no**, as appropriate.

Since backing up a disk can involve a lot of time and effort, **ufsdump** checkpoints at the start of each volume. If writing that volume fails for some reason, **ufsdump** will, with operator permission, restart itself from the checkpoint after a defective volume has been replaced.

**Suggested Dump
Schedule**

It is vital to perform full, "level 0", dumps at regular intervals. When performing a full dump, bring the machine down to single-user mode using **shutdown(1M)**. While preparing for a full dump, it is a good idea to clean the tape drive and heads. Incremental dumps should be performed with the system running in single-user mode.

Incremental dumps allow for convenient backup and recovery of active files on a more frequent basis, with a minimum of media and time. However, there are some tradeoffs. First, the interval between backups should be kept to a minimum (once a day at least). To guard against data loss as a result of a media failure (a rare, but possible occurrence), capture active files on (at least) two sets of dump volumes. Another consideration is the desire to keep unnecessary duplication of files to a minimum to save both operator time and media storage. A third consideration is the ease with which a particular backed-up version of a file can be located and restored. The following four-week schedule offers a reasonable tradeoff between these goals.

	<i>Sun</i>	<i>Mon</i>	<i>Tue</i>	<i>Wed</i>	<i>Thu</i>	<i>Fri</i>
<i>Week 1:</i>	Full	5	5	5	5	3
<i>Week 2:</i>		5	5	5	5	3
<i>Week 3:</i>		5	5	5	5	3
<i>Week 4:</i>		5	5	5	5	3

Although the Tuesday through Friday incrementals contain "extra copies" of files from Monday, this scheme assures that any file modified during the week can be recovered from the previous day's incremental dump.

**Process Priority of
ufsdump**

ufsdump uses multiple processes to allow it to read from the disk and write to the media concurrently. Due to the way it synchronizes between these processes, any attempt to run dump with a **nice** (process priority) of '-5' or better will likely make **ufsdump** run *slower* instead of faster.

**Overlapping
Partitions**

Most disks contain one or more overlapping slices because slice 2 covers the entire disk. The other slices are of various sizes and usually do not overlap. For example, a common configuration places **root** on slice 0, **swap** on slice 1, **/opt** on slice 5 and **/usr** on slice 6.

It should be emphasized that **ufsdump** dumps one **ufs** file system at a time. Given the above scenario where slice 0 and slice 2 have the same starting offset, executing **ufsdump** on slice 2 with the intent of dumping the entire disk would instead dump only the **root**

file system on slice 0. To dump the entire disk, the user must dump the file systems on each slice separately.

BUGS The `/etc/vfstab` file does not allow the desired frequency of backup for file systems to be specified (as `/etc/fstab` did). Consequently, the `w` and `W` options assume file systems should be backed up daily, which limits the usefulness of these options.

NAME	ufsrestore – incremental file system restore
SYNOPSIS	<code>/usr/lib/fs/ufs/ufsrestore options [arguments] [filename ...]</code>
DESCRIPTION	<p>ufsrestore restores files from backup media created with the ufsdump command. <i>options</i> is a single string of one-letter options. One (and only one) of the following options must be included: i, r, R, t, or x. <i>arguments</i> is one or more strings following <i>options</i>. The association of arguments with options is determined by order. That is, the first argument goes with the first option that takes an argument; the second argument goes with the second option that takes an argument, and so on. However, the <i>filename</i> arguments, which go with either the x or t options, must come last. They specify the names of files (or directories whose files) are to be restored to disk. Unless the h modifier is also used, a directory name refers to the files it contains, and (recursively) its subdirectories and the files they contain.</p>
OPTIONS Choose One	<p>One of the following options is required:</p> <ul style="list-style-type: none">i Interactive. After reading in the directory information from the media, ufsrestore invokes an interactive interface that allows you to browse through the dump file's directory hierarchy and select individual files to be extracted. See Interactive Commands, below, for a description of available commands.r Recursive. Restore the entire contents of the media into the current directory (which should be the top-level of the file system). To completely restore a file system, use this option to restore the level 0 dump, and again for each incremental dump. Although, this option is intended for a complete restore onto a clear file system, if the file system contains files not on the media, they are preserved.R Resume restoring. ufsrestore requests a particular volume of a multi-volume set from which to resume a full restore (see the r option above). This allows ufsrestore to start from a checkpoint when it is interrupted in the middle of a full restore.t Table of contents. List each <i>filename</i> that appears on the media. If no <i>filename</i> argument is given, the root directory is listed. This results in a list of all files on the media, unless the h modifier is in effect. The table of contents is taken from the media or from the specified archive file, when the a option is used. This option is mutually exclusive with the x and r options.x Extract the named files from the media. If a named file matches a directory whose contents were written onto the media, and the h modifier is not in effect, the directory is recursively extracted. The owner, modification time, and mode are restored (if possible). Existing files are overwritten and a warning is given. If no <i>filename</i> argument is given, the root directory is extracted. This results in the entire tape being extracted unless the h modifier is in effect.

Choose Any

In addition to one of the above options, any of the following options may be used:

a *archive_file*

Read the table of contents from *archive_file* instead of the media. This option can be used in combination with the **t**, **i**, or **x** options, making it possible to check whether files are on the media without having to mount the media. When used with the **x** and interactive (**i**) options, it prompts for the volume containing the file(s) before extracting them.

b factor Blocking factor. Specify the blocking factor for tape reads. For variable length SCSI tape devices, unless the data was written with the default blocking factor, a blocking factor at least as great as that used to write the tape must be used; otherwise, an error will be generated. Note that a tape block is 512 bytes. Refer to the man page for your specific tape driver for the maximum blocking factor.

c Convert the contents of the media in 4.1BSD format to the new **ufs** file system format.

d Debug. Turn on debugging output.

f *dump_file*

Use *dump_file* instead of **/dev/rmt/0** as the file to restore from. Typically *dump_file* specifies a tape or diskette drive. If *dump_file* is specified as **'-'**, **ufsrestore** reads from the standard input. This allows, **ufsdump(1M)** and **ufsrestore** to be used in a pipeline to copy a file system:

```
example# ufsdump 0f - /dev/rdisk/c0t0d0s7 | (cd /home;ufsrestore xf -)
```

If the name of the file is of the form *machine:device*, the restore is done from the specified machine over the network using **rmt(1M)**. Since **ufsrestore** is normally run by root, the name of the local machine must appear in the **.rhosts** file of the remote machine. If the file is specified as *user@machine:device*, **ufsrestore** will attempt to execute as the specified user on the remote machine. The specified user must have a **.rhosts** file on the remote machine that allows the user invoking the command from the local machine to access the remote machine.

h Extract or list the actual directory, rather than the files that it references. This prevents hierarchical restoration of complete subtrees from the tape.

m Extract by inode numbers rather than by filename to avoid regenerating complete pathnames. Regardless of where the files are located in the dump hierarchy, they are restored into the current directory and renamed with their inode number. This is useful if only a few files are being extracted.

s n Skip to the *n*'th file when there are multiple dump files on the same tape. For example, the command:

```
example# ufsrestore xfs /dev/rmt/0hn 5
```

would position you at the fifth file on the tape.

v Verbose. **ufsrestore** displays the name and inode number of each file it restores, preceded by its file type.

**Interactive
Commands**

y Do not ask whether to abort the restore in the event of tape errors. **ufsrestore** tries to skip over the bad tape block(s) and continue as best it can.

ufsrestore enters interactive mode when invoked with the **i** option. Interactive commands are reminiscent of the shell. For those commands that accept an argument, the default is the current directory. The interactive options are:

add [*filename*]

Add the named file or directory to the list of files to extract. If a directory is specified, add that directory and its files (recursively) to the extraction list (unless the **h** modifier is in effect).

cd *directory*

Change to *directory* (within the dump file).

delete [*filename*]

Delete the current directory, or the named file or directory from the list of files to extract. If a directory is specified, delete that directory and all its descendents from the extraction list (unless the **h** modifier is in effect). The most expedient way to extract a majority of files from a directory is to add that directory to the extraction list, and then delete specific files to omit.

extract Extract all files on the extraction list from the dump media. **ufsrestore** asks which volume the user wishes to mount. The fastest way to extract a small number of files is to start with the last volume and work toward the first.

help Display a summary of the available commands.

ls [*directory*]

List files in *directory* or the current directory, represented by a '.' (period). Directories are appended with a '/' (slash). Entries marked for extraction are prefixed with a '*' (asterisk). If the verbose option is in effect, inode numbers are also listed.

pwd Print the full pathname of the current working directory.

quit **ufsrestore** exits immediately, even if the extraction list is not empty.

setmodes

Prompts: **set owner/mode for** '.' (period). Type **y** for yes to set the mode (permissions, owner, times) of the current directory '.' (period) into which files are being restored equal to the mode of the root directory of the file system from which they were dumped. Normally, this is what you want when restoring a whole file system, or restoring individual files into the same locations from which they were dumped. Type **n** for no, to leave the mode of the current directory unchanged. Normally, this is what you want when restoring part of a dump to a directory other than the one from which the files were dumped.

verbose

Toggle the status of the **v** modifier. While **v** is in effect, the **ls** command lists the inode numbers of all entries, and **ufsrestore** displays information about each file as it is extracted.

what Display the dump header on the media.

FILES	/dev/rmt/0	the default tape drive
	/tmp/rstdir*	file containing directories on the tape
	/tmp/rstmode*	owner, mode, and timestamps for directories
	./restoresymtable	information passed between incremental restores

SEE ALSO **mkfs(1M)**, **mount(1M)**, **rmt(1M)**, **ufsdump(1M)**

DIAGNOSTICS **ufsrestore** complains about bad option characters.

Read errors result in complaints. If **y** has been specified, or the user responds **y**, **ufsrestore** will attempt to continue.

If the dump extends over more than one tape, **ufsrestore** asks the user to change tapes. If the **x** or **i** option has been specified, **ufsrestore** also asks which volume the user wishes to mount.

There are numerous consistency checks that can be listed by **ufsrestore**. Most checks are self-explanatory or can “never happen”. Common errors are given below.

Converting to new file system format

A dump tape created from the old file system has been loaded. It is automatically converted to the new file system format.

filename: not found on tape

The specified file name was listed in the tape directory, but was not found on the tape. This is caused by tape read errors while looking for the file, or from using a dump tape created on an active file system.

expected next file *inumber*, got *inumber*

A file that was not listed in the directory showed up. This can occur when using a dump tape created on an active file system.

Incremental tape too low

When doing an incremental restore, a tape that was written before the previous incremental tape, or that has too low an incremental level has been loaded.

Incremental tape too high

When doing incremental restore, a tape that does not begin its coverage where the previous incremental tape left off, or one that has too high an incremental level has been loaded.

media read error: invalid argument

Blocking factor specified for read is smaller than the blocking factor used to write data.

Tape read error while restoring *filename*

Tape read error while skipping over inode *inumber*

Tape read error while trying to resynchronize

A tape read error has occurred

If a file name is specified, then its contents are probably partially wrong. If an inode is being skipped or the tape is trying to resynchronize, then no extracted

files have been corrupted, though files may not be found on the tape.

resync ufsrestore, skipped *num*

After a tape read error, **ufsrestore** may have to resynchronize itself. This message lists the number of blocks that were skipped over.

NOTES

ufsrestore can get confused when doing incremental restores from dump tapes that were made on active file systems.

A “level 0” dump must be done after a full restore. Because **ufsrestore** runs in user mode, it has no control over inode allocation. This means that **ufsrestore** repositions the files, although it does not change their contents. Thus, a full dump must be done to get a new set of directories reflecting the new file positions, so that later incremental dumps will be correct.

NAME	unshare – make local resource unavailable for mounting by remote systems
SYNOPSIS	unshare [-F <i>FSType</i>] [-o <i>specific_options</i>] [<i>pathname</i> <i>resourcename</i>]
AVAILABILITY	SUNWcsu
DESCRIPTION	The unshare command makes a shared local resource unavailable as file system type <i>FSType</i> . If the option -F <i>FSType</i> is omitted, then the first file system type listed in file <i>/etc/dfs/fstypes</i> will be used as the default. <i>Specific_options</i> , as well as the semantics of <i>resourcename</i> , are specific to particular distributed file systems.
OPTIONS	-F <i>FSType</i> Specify the file system type. -o <i>specific_options</i> Specify options specific to the file system provided by the -F option.
FILES	<i>/etc/dfs/fstypes</i> <i>/etc/dfs/sharetab</i>
SEE ALSO	share (1M), shareall (1M)
NOTES	If <i>pathname</i> or <i>resourcename</i> is not found in the shared information, an error message will be sent to standard error.

NAME	unshare_nfs – make local NFS filesystems unavailable for mounting by remote systems
SYNOPSIS	unshare [-F nfs] <i>pathname</i>
DESCRIPTION	The unshare command makes local filesystems unavailable for mounting by remote systems. The shared filesystem must correspond to a line with NFS as the <i>FSType</i> in the file /etc/dfs/sharetab .
OPTIONS	-F This option may be omitted if NFS is the first file system type listed in the file /etc/dfs/fstypes .
FILES	/etc/dfs/fstypes /etc/dfs/sharetab
SEE ALSO	share(1M)
NOTES	<p>If the filesystem being unshared is a symbolic link to a valid pathname, the canonical path (the path which the symbolic link follows) will be unshared.</p> <p>For example, if /export/foo is a symbolic link to /export/bar (/export/foo -> /export/bar), the following unshare command will result in /export/bar as the unshared pathname (and not /export/foo).</p> <p style="text-align: center;">example# unshare -F nfs /export/foo</p>

NAME	useradd – administer a new user login on the system
SYNOPSIS	<pre> useradd [<i>-c comment</i>] [<i>-d dir</i>] [<i>-e expire</i>] [<i>-f inactive</i>] [<i>-g group</i>] [<i>-G group [, group ...]</i>] [<i>-m [-k skel_dir]</i>] [<i>-u uid [-o]</i>] [<i>-s shell</i>] <i>login</i> useradd -D [<i>-b base_dir</i>] [<i>-e expire</i>] [<i>-f inactive</i>] [<i>-g group</i>] </pre>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>useradd adds a new user entry to the /etc/passwd and /etc/shadow files. It also creates supplementary group memberships for the user (-G option) and creates the home directory (-m option) for the user if requested. The new login remains locked until the passwd(1) command is executed.</p> <p>Specifying useradd -D with the -g, -b, -f, or -e options (or any combination of these) sets the default values for the respective fields. See the -D option below. Subsequent useradd commands without the -D option use these arguments.</p> <p>The system file entries created with this command have a limit of 512 characters per line. Specifying long arguments to several options may exceed this limit.</p> <p><i>login</i> is a string of printable characters that specifies the new login name of the user. It may not contain a colon (:) or a newline (\n).</p>
OPTIONS	<p>-c <i>comment</i> Any text string. It is generally a short description of the login, and is currently used as the field for the user's full name. This information is stored in the user's /etc/passwd entry.</p> <p>-d <i>dir</i> The home directory of the new user. It defaults to <i>base_dir/login</i>, where <i>base_dir</i> is the base directory for new login home directories and <i>login</i> is the new login name.</p> <p>-e <i>expire</i> Specify the expiration date for a login. After this date, no user will be able to access this login. <i>expire</i> is a date entered in any format you like (except a Julian date). If the date format that you choose includes spaces, it must be quoted. For example, you may enter 10/6/90 or "October 6, 1990". A null value (" ") defeats the status of the expired date. This option is useful for creating temporary logins.</p> <p>-f <i>inactive</i> The maximum number of days allowed between uses of a login ID before that login ID is declared invalid. Normal values are positive integers. A value of 0 defeats the status.</p> <p>-g <i>group</i> An existing group's integer ID or character-string name. Without the -D option, it defines the new user's primary group membership and defaults to the default group. You can reset this default value by invoking useradd -D -g <i>group</i>.</p>

- G** *group* An existing group's integer ID or character-string name. It defines the new user's supplementary group membership. Duplicates between *group* with the **-g** and **-G** options are ignored. No more than NGROUPS_MAX groups may be specified.
- k** *skel_dir* A directory that contains skeleton information (such as **.profile**) that can be copied into a new user's home directory. This directory must already exist. The system provides the **/etc/skel** directory that can be used for this purpose.
- m** Create the new user's home directory if it does not already exist. If the directory already exists, it must have read, write, and execute permissions by *group*, where *group* is the user's primary group.
- s** *shell* Full pathname of the program used as the user's shell on login. It defaults to an empty field causing the system to use **/sbin/sh** as the default. The value of *shell* must be a valid executable file.
- u** *uid* The UID of the new user. This UID must be a non-negative decimal integer below MAXUID as defined in **<sys/param.h>**. The UID defaults to the next available (unique) number above the highest number currently assigned. For example, if UIDs 100, 105, and 200 are assigned, the next default UID number will be 201. (UIDs from 0-99 are reserved by SunOS for future applications.)
- o** This option allows a UID to be duplicated (non-unique).
- D** Display the default values for *group*, *base_dir*, *skel_dir*, *shell*, *inactive*, and *expire*. When used with the **-g**, **-b**, **-f**, or **-e** options, the **-D** option sets the default values for the specified fields. The default values are:
- | | |
|-----------------|-------------------------|
| <i>group</i> | other (GID of 1) |
| <i>base_dir</i> | /home |
| <i>skel_dir</i> | /etc/skel |
| <i>shell</i> | /sbin/sh |
| <i>inactive</i> | 0 |
| <i>expire</i> | Null (unset). |
- b** *base_dir* The default base directory for the system if **-d** *dir* is not specified. *base_dir* is concatenated with the user's login to define the home directory. If the **-m** option is not used, *base_dir* must exist.

FILES

/etc/passwd
/etc/shadow
/etc/group
/etc/skel

SEE ALSO **passwd(1)**, **users(1B)**, **groupadd(1M)**, **groupdel(1M)**, **groupmod(1M)**, **logins(1M)**, **userdel(1M)**, **usermod(1M)**, **passwd(4)**

DIAGNOSTICS

In case of an error, **useradd** prints an error message and exits with a non-zero status.

UX: useradd: ERROR: login is already in use. Choose another.

The *login* specified is already in use.

UX: useradd: ERROR: uid *uid* is already in use. Choose another.

The *uid* specified with the **-u** option is not unique.

UX: useradd: ERROR: group *group* does not exist. Choose another.

The *group* specified with the **-g** option is already in use.

UX: useradd: WARNING: uid *uid* is reserved.

The *uid* specified with the **-u** option is in the range of reserved UIDs (from 0-99).

UX: useradd: ERROR: uid *uid* is too big. Choose another.

The *uid* specified with the **-u** option exceeds **MAXUID** as defined in **<sys/param.h>**.

UX: useradd: ERROR: Cannot update system files - login cannot be created.

The **/etc/passwd** or **/etc/shadow** files do not exist.

NOTES

useradd only adds a user definition to the local system. If a network nameservice such as NIS or NIS+ is being used to supplement the local **/etc/passwd** file with additional entries, **useradd** cannot change information supplied by the network nameservice. However **useradd** will verify the uniqueness of the user name and user id and the existence of any group names specified against the external nameservice.

NAME	userdel – delete a user’s login from the system
SYNOPSIS	userdel [-r] <i>login</i>
AVAILABILITY	SUNWcsu
DESCRIPTION	The userdel command deletes a user’s login from the system and makes the appropriate login-related changes to the system file and file system.
OPTIONS	<p>-r Remove the user’s home directory from the system. This directory must exist. The files and directories under the home directory will no longer be accessible following successful execution of the command.</p> <p><i>login</i> A string of printable characters that specify an existing login on the system. It may not contain a colon (:) or a newline (\n).</p>
FILES	<p>/etc/passwd</p> <p>/etc/shadow</p> <p>/etc/group</p>
SEE ALSO	passwd(1) , users(1B) , groupadd(1M) , groupdel(1M) , groupmod(1M) , logins(1M) , useradd(1M) , usermod(1M) , passwd(4)
DIAGNOSTICS	<p>The userdel command exits with one of the following values:</p> <p>0 Success.</p> <p>2 Invalid command syntax. A usage message for the userdel command is displayed.</p> <p>6 The login to be removed does not exist.</p> <p>8 The login to be removed is in use.</p> <p>10 Cannot update the /etc/group file but the login is removed from the /etc/passwd file.</p> <p>12 Cannot remove or otherwise modify the home directory.</p>
NOTES	userdel only deletes a user definition that is in the local /etc/passwd file. If a network name service such as NIS or NIS+ is being used to supplement the local /etc/passwd file with additional entries, userdel cannot change information supplied by the network name service.

NAME	usermod – modify a user’s login information on the system
SYNOPSIS	usermod [-u <i>uid</i> [-o]] [-g <i>group</i>] [-G <i>group</i> [, <i>group</i> . . .]] [-d <i>dir</i> [-m]] [-s <i>shell</i>] [-c <i>comment</i>] [-l <i>new_logname</i>] [-f <i>inactive</i>] [-e <i>expire</i>] <i>login</i>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>usermod modifies a user’s login definition on the system. It changes the definition of the specified login and makes the appropriate login-related system file and file system changes.</p> <p>The system file entries created with this command have a limit of 512 characters per line. Specifying long arguments to several options may exceed this limit.</p> <p><i>login</i> is a string of printable characters that specifies the existing login name of the user. It must exist and may not contain a colon (:) or a newline (\n).</p>
OPTIONS	<p>-u <i>uid</i> Specify a new UID for the user. It must be a non-negative decimal integer less than MAXUID as defined in <param.h>. Note that the UID associated with the user’s home directory is not modified with this option; a user will not have access to their home directory until the UID is manually reassigned using chown(1M).</p> <p>-o This option allows the specified UID to be duplicated (non-unique).</p> <p>-g <i>group</i> Specify an existing group’s integer ID or character-string name. It redefines the user’s primary group membership.</p> <p>-G <i>group</i> Specify an existing group’s integer "ID" ", " or character string name. It redefines the user’s supplementary group membership. Duplicates between <i>group</i> with the -g and -G options are ignored. No more than NGROUPS_UMAX groups may be specified as defined in <param.h>.</p> <p>-d <i>dir</i> Specify the new home directory of the user. It defaults to <i>base_dir/login</i>, where <i>base_dir</i> is the base directory for new login home directories, and <i>login</i> is the new login.</p> <p>-m Move the user’s home directory to the new directory specified with the -d option. If the directory already exists, it must have permissions read/write/execute by <i>group</i>, where <i>group</i> is the user’s primary group.</p> <p>-s <i>shell</i> Specify the full pathname of the program that is used as the user’s shell on login. The value of <i>shell</i> must be a valid executable file.</p> <p>-c <i>comment</i> Specify a comment string. <i>comment</i> can be any text string. It is generally a short description of the login, and is currently used as the field for the user’s full name. This information is stored in the user’s /etc/passwd entry.</p> <p>-l <i>new_logname</i> Specify a string of printable characters that specifies the new login name for the user. It may not contain a colon (:) or a newline (\n).</p> <p>-e <i>expire</i> Specify the future date on which a login can no longer be used; after this</p>

date, no user will be able to access this login. This option is useful for creating temporary logins. You may type the value of the argument *expire* (which is a date) in any format you like (except a Julian date). For example, you may enter **10/6/90** or **October 6, 1990**. A value of “ ” defeats the status of the expired date.

-f inactive Specify the maximum number of days allowed between uses of a login ID before that login ID is declared invalid. Normal values are positive integers. A value of **0** defeats the status.

EXIT CODES

In case of an error, **usermod** prints an error message and exits with one of the following values:

- 2** The command syntax was invalid. A usage message for the **usermod** command is displayed.
- 3** An invalid argument was provided to an option.
- 4** The *uid* given with the **-u** option is already in use.
- 5** The password files contain an error. **pwconv(1M)** can be used to correct possible errors. See **passwd(4)**.
- 6** The login to be modified does not exist, the *group* does not exist, or the login shell does not exist.
- 8** The login to be modified is in use.
- 9** The *new_logname* is already in use.
- 10** Cannot update the **/etc/group** file. Other update requests will be implemented.
- 11** Insufficient space to move the home directory (**-m** option). Other update requests will be implemented.
- 12** Unable to complete the move of the home directory to the new home directory.

FILES

/etc/passwd	system password file
/etc/shadow	system file containing users' encrypted passwords and related information
/etc/group	system file containing group definitions

SEE ALSO

passwd(1), **users(1B)**, **chown(1M)**, **groupadd(1M)**, **groupdel(1M)**, **groupmod(1M)**, **logins(1M)**, **pwconv(1M)**, **useradd(1M)**, **userdel(1M)**, **passwd(4)**

NOTES

usermod only modifies **passwd** definitions in the local **/etc/passwd** and **/etc/shadow** files. If a network nameservice such as NIS or NIS+ is being used to supplement the local files with additional entries, **usermod** cannot change information supplied by the network nameservice. However **usermod** will verify the uniqueness of user name and user ID against the external nameservice.

NAME	uuccheck – check the uucp directories and permissions file
SYNOPSIS	/usr/lib/uucp/uuccheck [-v] [-xdebug-level]
DESCRIPTION	uuccheck checks for the presence of the uucp system required files and directories. uuccheck also does error checking of the Permissions file (/etc/uucp/Permissions). uuccheck is executed during package installation. Note that uuccheck can only be used by the super-user or uucp .
OPTIONS	-v Give a detailed (verbose) explanation of how the uucp programs will interpret the Permissions file. -xdebug-level Produce debugging output on the standard output. <i>debug-level</i> is a number from 0 to 9 . Higher numbers give more detailed debugging information.
FILES	/etc/uucp/Devices /etc/uucp/Limits /etc/uucp/Permissions /etc/uucp/Systems /var/spool/locks/* /var/spool/uucp/* /var/spool/uucppublic/*
SEE ALSO	uucp(1C) , uustat(1C) , uux(1C) , uucico(1M) , uusched(1M)
BUGS	The program does not check file/directory modes or some errors in the Permissions file such as duplicate login or machine name.

NAME	uucico – file transport program for the uucp system
SYNOPSIS	<code>/usr/lib/uucp/uucico [-f] [-ctype] [-dspool-directory] [-iinterface] [-rrole-number] [-ssystem-name] [-xdebug-level]</code>
DESCRIPTION	uucico is the file transport program for uucp work file transfers.
OPTIONS	<p>-f This option is used to "force execution" of uucico by ignoring the limit on the maximum number of uucicos defined in the /etc/uucp/Limits file.</p> <p>-ctype The first field in the Devices file is the "Type" field. The -c option forces uucico to only use entries in the "Type" field that match the user specified <i>type</i>. The specified <i>type</i> is usually the name of a local area network.</p> <p>-dspool-directory This option specifies the directory <i>spool-directory</i> that contains the uucp work files to be transferred. The default spool directory is /var/spool/uucp.</p> <p>-iinterface This option defines the <i>interface</i> used with uucico. The interface only affects slave mode. Known interfaces are UNIX (default), TLI (basic Transport Layer Interface), and TLIS (Transport Layer Interface with Streams modules, read/write).</p> <p>-rrole-number The <i>role-number 1</i> is used for master mode. <i>role-number 0</i> is used for slave mode (default). When uucico is started by a program or cron, <i>role-number 1</i> should be used for master mode.</p> <p>-ssystem-name The -s option defines the remote system (<i>system-name</i>) that uucico will try to contact. It is required when the role is master; <i>system-name</i> must be defined in the Systems file.</p> <p>-xdebug-level Both uux and uucp queue jobs that will be transferred by uucico. These jobs are normally started by the uusched scheduler, for debugging purposes, and can be started manually. For example, the shell Uutry starts uucico with debugging turned on. The <i>debug-level</i> is a number between 0 and 9. Higher numbers give more detailed debugging information.</p>
FILES	/etc/uucp/Devconfig /etc/uucp/Devices /etc/uucp/Limits /etc/uucp/Permissions /etc/uucp/Sysfiles /etc/uucp/Systems /var/spool/locks/* /var/spool/uucp/* /var/spool/uucppublic/*

SEE ALSO

uucp(1C), uustat(1C), uux(1C), Uutry(1M), cron(1M), uusched(1M)

NAME	uucleanup – uucp spool directory clean-up
SYNOPSIS	/usr/lib/uucp/uucleanup [-Ctime] [-Dtime] [-mstring] [-otime] [-ssystem] [-Wtime] [-xdebug-level] [-Xtime]
DESCRIPTION	<p>uucleanup will scan the spool directories for old files and take appropriate action to remove them in a useful way:</p> <ul style="list-style-type: none"> • Inform the requester of send/receive requests for systems that can not be reached. • Return undeliverable mail to the sender. • Deliver rnews files addressed to the local system. • Remove all other files. <p>In addition, there is a provision to warn users of requests that have been waiting for a given number of days (default 1 day). Note: uucleanup will process as if all option <i>times</i> were specified to the default values unless <i>time</i> is specifically set.</p> <p>This program is typically started by the shell uudemon.cleanup, which should be started by cron(1M).</p>
OPTIONS	<p>-Ctime Remove any C. files greater or equal to <i>time</i> days old and send appropriate information to the requester (default 7 days).</p> <p>-Dtime Remove any D. files greater or equal to <i>time</i> days old, make an attempt to deliver mail messages, and execute rnews when appropriate (default 7 days).</p> <p>-mstring Include <i>string</i> in the warning message generated by the -W option. The default line is "See your local administrator to locate the problem".</p> <p>-otime Delete other files whose age is more than <i>time</i> days (default 2 days).</p> <p>-ssystem Execute for <i>system</i> spool directory only.</p> <p>-Wtime Any C. files equal to <i>time</i> days old will cause a mail message to be sent to the requester warning about the delay in contacting the remote. The message includes the <i>JOBID</i>, and in the case of mail, the mail message. The administrator may include a message line telling whom to call to check the problem (-m option) (default 1 day).</p> <p>-xdebug-level Produce debugging output on standard output. <i>debug-level</i> is a single digit between 0 and 9; higher numbers give more detailed debugging information. (This option may not be available on all systems.)</p> <p>-Xtime Any X. files greater or equal to <i>time</i> days old will be removed. The D. files are probably not present (if they were, the X. could get executed). But if there are D. files, they will be taken care of by D. processing (default 2 days).</p>

FILES **/usr/lib/uucp** directory with commands used by **uucleanup** internally
 /var/spool/uucp spool directory

SEE ALSO **uucp(1C), uux(1C), cron(1M)**

NAME	uusched – the scheduler for the uucp file transport program
SYNOPSIS	/usr/lib/uucp/uusched [<i>-udebug-level</i>] [<i>-xdebug-level</i>]
DESCRIPTION	uusched is the uucp (1C) file transport scheduler. It is usually started by the daemon <i>uudemon.hour</i> that is started by cron (1M) from an entry in user uucp 's crontab file: 11,41 * * * * /etc/uucp/uucp/uudemons.hour
OPTIONS	The options are for debugging purposes only. <i>debug-level</i> is a number between 0 and 9. Higher numbers give more detailed debugging information: <i>-udebug-level</i> The <i>-udebug-level</i> option is passed to uucico (1M) as <i>-xdebug-level</i> . <i>-xdebug-level</i> Outputs debugging messages from uusched .
FILES	/etc/uucp/Devices /etc/uucp/Permissions /etc/uucp/Systems /var/spool/locks/* /var/spool/uucp/* /var/spool/uucppublic/*
SEE ALSO	uucp (1C), uustat (1C), uux (1C), cron (1M), uucico (1M)

NAME	uuxqt – execute remote command requests
SYNOPSIS	/usr/lib/uucp/uuxqt [<i>-ssystem</i>] [<i>-xdebug-level</i>]
DESCRIPTION	<p>uuxqt is the program that executes remote job requests from remote systems generated by the use of the uux command. (mail uses uux for remote mail requests). uuxqt searches the spool directories looking for execution requests. For each request, uuxqt checks to see if all the required data files are available, accessible, and the requested commands are permitted for the requesting system. The Permissions file is used to validate file accessibility and command execution permission.</p> <p>There are two environment variables that are set before the uuxqt command is executed:</p> <ul style="list-style-type: none">• UU_MACHINE is the machine that sent the job (the previous one).• UU_USER is the user that sent the job. <p>These can be used in writing commands that remote systems can execute to provide information, auditing, or restrictions.</p>
OPTIONS	<p><i>-ssystem</i> Specifies the remote <i>system</i> name.</p> <p><i>-xdebug-level</i> <i>debug-level</i> is a number from 0 to 9. Higher numbers give more detailed debugging information.</p>
FILES	<p>/etc/uucp/Limits /etc/uucp/Permissions /var/spool/locks/* /var/spool/uucp/*</p>
SEE ALSO	mail(1), uucp(1C), uustat(1C), uux(1C), uucico(1M)

NAME	vmstat – report virtual memory statistics																																																																																																																																																																																											
SYNOPSIS	vmstat [-cisS] [<i>disks</i>] [<i>interval</i> [<i>count</i>]]																																																																																																																																																																																											
DESCRIPTION	<p>vmstat delves into the system and reports certain statistics kept about process, virtual memory, disk, trap and CPU activity. Note: vmstat statistics are only supported for certain devices.</p> <p>Without options, vmstat displays a one-line summary of the virtual memory activity since the system was booted. If <i>interval</i> is specified, vmstat summarizes activity over the last <i>interval</i> seconds, repeating forever. If a <i>count</i> is given, the statistics are repeated <i>count</i> times. Note: <i>interval</i> and <i>count</i> do not apply to the -i and -s options. If <i>disks</i> are specified, they are given priority when vmstat chooses which disks to display (only four fit on a line). Common disk names are id, sd, xd, or xy, followed by a number, for example, sd2, xd0, and so forth. For more general system statistics, use sar(1), iostat(1M), or sar(1M).</p> <p>See <i>Solaris 1.x to Solaris 2.x Transition Guide</i> for device naming conventions for disks.</p>																																																																																																																																																																																											
OPTIONS	<p>-c Report cache flushing statistics. By default, report the total number of each kind of cache flushed since boot time. The types are: user, context, region, segment, page, and partial-page.</p> <p>-i Report the number of interrupts per device.</p> <p>-s Display the total number of various system events since boot.</p> <p>-S Report on swapping rather than paging activity. This option will change two fields in vmstat's "paging" display: rather than the "re" and "mf" fields, vmstat will report "si" (swap-ins) and "so" (swap-outs).</p>																																																																																																																																																																																											
EXAMPLES	<p>The following command displays a summary of what the system is doing every five seconds.</p> <pre>example% vmstat 5</pre> <table border="0"> <thead> <tr> <th>procs</th> <th>memory</th> <th colspan="4">page</th> <th colspan="4">disk</th> <th colspan="3">faults</th> <th colspan="3">cpu</th> </tr> <tr> <th>r b w</th> <th>swap free</th> <th>re</th> <th>mf</th> <th>pi</th> <th>po</th> <th>fr</th> <th>de</th> <th>sr</th> <th>s0</th> <th>s1</th> <th>s2</th> <th>s3</th> <th>in</th> <th>sy</th> <th>cs</th> <th>us</th> <th>sy</th> <th>id</th> </tr> </thead> <tbody> <tr> <td>0 0 0</td> <td>11456 4120</td> <td>1</td> <td>41</td> <td>19</td> <td>1</td> <td>3</td> <td>0</td> <td>2</td> <td>0</td> <td>4</td> <td>0</td> <td>0</td> <td>48</td> <td>112</td> <td>130</td> <td>4</td> <td>14</td> <td>82</td> </tr> <tr> <td>0 0 1</td> <td>10132 4280</td> <td>0</td> <td>4</td> <td>44</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>23</td> <td>0</td> <td>0</td> <td>211</td> <td>230</td> <td>144</td> <td>3</td> <td>35</td> <td>62</td> </tr> <tr> <td>0 0 1</td> <td>10132 4616</td> <td>0</td> <td>0</td> <td>20</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>19</td> <td>0</td> <td>0</td> <td>150</td> <td>172</td> <td>146</td> <td>3</td> <td>33</td> <td>64</td> </tr> <tr> <td>0 0 1</td> <td>10132 5292</td> <td>0</td> <td>0</td> <td>9</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>21</td> <td>0</td> <td>0</td> <td>165</td> <td>105</td> <td>130</td> <td>1</td> <td>21</td> <td>78</td> </tr> <tr> <td>1 1 1</td> <td>10132 5496</td> <td>0</td> <td>0</td> <td>5</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>23</td> <td>0</td> <td>0</td> <td>183</td> <td>92</td> <td>134</td> <td>1</td> <td>20</td> <td>79</td> </tr> <tr> <td>1 0 1</td> <td>10132 5564</td> <td>0</td> <td>0</td> <td>25</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>18</td> <td>0</td> <td>0</td> <td>131</td> <td>231</td> <td>116</td> <td>4</td> <td>34</td> <td>62</td> </tr> <tr> <td>1 0 1</td> <td>10124 5412</td> <td>0</td> <td>0</td> <td>37</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>22</td> <td>0</td> <td>0</td> <td>166</td> <td>179</td> <td>118</td> <td>1</td> <td>33</td> <td>67</td> </tr> <tr> <td>1 0 1</td> <td>10124 5236</td> <td>0</td> <td>0</td> <td>24</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>14</td> <td>0</td> <td>0</td> <td>109</td> <td>243</td> <td>113</td> <td>4</td> <td>56</td> <td>39</td> </tr> </tbody> </table> <pre>^C example%</pre>	procs	memory	page				disk				faults			cpu			r b w	swap free	re	mf	pi	po	fr	de	sr	s0	s1	s2	s3	in	sy	cs	us	sy	id	0 0 0	11456 4120	1	41	19	1	3	0	2	0	4	0	0	48	112	130	4	14	82	0 0 1	10132 4280	0	4	44	0	0	0	0	0	23	0	0	211	230	144	3	35	62	0 0 1	10132 4616	0	0	20	0	0	0	0	0	19	0	0	150	172	146	3	33	64	0 0 1	10132 5292	0	0	9	0	0	0	0	0	21	0	0	165	105	130	1	21	78	1 1 1	10132 5496	0	0	5	0	0	0	0	0	23	0	0	183	92	134	1	20	79	1 0 1	10132 5564	0	0	25	0	0	0	0	0	18	0	0	131	231	116	4	34	62	1 0 1	10124 5412	0	0	37	0	0	0	0	0	22	0	0	166	179	118	1	33	67	1 0 1	10124 5236	0	0	24	0	0	0	0	0	14	0	0	109	243	113	4	56	39
procs	memory	page				disk				faults			cpu																																																																																																																																																																															
r b w	swap free	re	mf	pi	po	fr	de	sr	s0	s1	s2	s3	in	sy	cs	us	sy	id																																																																																																																																																																										
0 0 0	11456 4120	1	41	19	1	3	0	2	0	4	0	0	48	112	130	4	14	82																																																																																																																																																																										
0 0 1	10132 4280	0	4	44	0	0	0	0	0	23	0	0	211	230	144	3	35	62																																																																																																																																																																										
0 0 1	10132 4616	0	0	20	0	0	0	0	0	19	0	0	150	172	146	3	33	64																																																																																																																																																																										
0 0 1	10132 5292	0	0	9	0	0	0	0	0	21	0	0	165	105	130	1	21	78																																																																																																																																																																										
1 1 1	10132 5496	0	0	5	0	0	0	0	0	23	0	0	183	92	134	1	20	79																																																																																																																																																																										
1 0 1	10132 5564	0	0	25	0	0	0	0	0	18	0	0	131	231	116	4	34	62																																																																																																																																																																										
1 0 1	10124 5412	0	0	37	0	0	0	0	0	22	0	0	166	179	118	1	33	67																																																																																																																																																																										
1 0 1	10124 5236	0	0	24	0	0	0	0	0	14	0	0	109	243	113	4	56	39																																																																																																																																																																										

The fields of **vmstat**'s display are:

- procs** Report the number of processes in each of the three following states:
r in run queue
b blocked for resources (I/O, paging, and so forth)
w runnable but swapped
- memory** Report on usage of virtual and real memory.
swap amount of swap space currently available (Kbytes)
free size of the free list (Kbytes)
- page** Report information about page faults and paging activity. The information on each of the following activities is given in units per second.
re page reclaims — but see the **-S** option for how this field is modified.
mf minor faults — but see the **-S** option for how this field is modified.
pi kilobytes paged in
po kilobytes paged out
fr kilobytes freed
de anticipated short-term memory shortfall (Kbytes)
sr pages scanned by clock algorithm
- disk** Report the number of disk operations per second. There are slots for up to four disks, labeled with a single letter and number. The letter indicates the type of disk (s = SCSI, i = IPI, and so forth); the number is the logical unit number.
- faults** Report the trap/interrupt rates (per second).
in (non clock) device interrupts
sy system calls
cs CPU context switches
- cpu** Give a breakdown of percentage usage of CPU time. On MP systems, this is an average across all processors.
us user time
sy system time
id idle time

SEE ALSO

sar(1), **iostat(1M)**, **sar(1M)**

System Administration Guide, Volume II

NAME	volcopy – make an image copy of file system
SYNOPSIS	volcopy [-F <i>FSType</i>] [-V] [<i>generic_options</i>] [-o <i>FSType-specific_options</i>] <i>operands</i>
AVAILABILITY	SUNWcsu
DESCRIPTION	volcopy makes a literal copy of the file system.
OPTIONS	<p>The following options are supported:</p> <p>-F <i>FSType</i> Specify the <i>FSType</i> on which to operate. The <i>FSType</i> should either be specified here or be determinable from /etc/vfstab by matching the <i>operands</i> with an entry in the table. Otherwise, the default file system type specified in /etc/default/fs will be used.</p> <p>-V Echo the complete command line, but do not execute the command. The command line is generated by using the options and arguments provided by the user and adding to them information derived from /etc/vfstab. This option should be used to verify and validate the command line.</p> <p><i>generic_options</i> Options that are commonly supported by most <i>FSType</i>-specific command modules. The following options are available:</p> <p> -a Require the operator to respond “yes” or “no” instead of simply waiting ten seconds before the copy is made.</p> <p> -s (Default) Invoke the DEL if wrong verification sequence.</p> <p>-o <i>FSType-specific_options</i> Specify <i>FSType</i>-specific options in a comma separated (without spaces) list of suboptions and keyword-attribute pairs for interpretation by the <i>FSType</i>-specific module of the command.</p>
OPERANDS	<p>The following operand is supported:</p> <p><i>operands</i> generally include the device and volume names and are file system specific. A detailed description of the <i>operands</i> can be found on the <i>FSType</i>-specific man pages of volcopy.</p>
EXIT STATUS	<p>The following exit values are returned:</p> <p>0 Successful file system copy</p> <p>1 An error has occurred.</p>
FILES	<p>/etc/vfstab list of default parameters for each file system</p> <p>/etc/default/fs default local file system type. Default values can be set for the following flags in /etc/default/fs. For example: LOCAL=ufs.</p> <p>LOCAL: The default partition for a command if no <i>FSType</i> is specified.</p>

SEE ALSO **labelit(1M)**, **vfstab(4)**
Manual pages for the *FSType*-specific modules of **volcopy**.

NOTES This command may not be supported for all *FSTypes*.

NAME	volcopy_ufs – make an image copy of a ufs file system
SYNOPSIS	volcopy [-F ufs] [<i>generic_options</i>] <i>fsname srcdevice volname1 destdevice volname2</i>
DESCRIPTION	volcopy makes a literal copy of the ufs file system using a blocksize matched to the device.
OPTIONS	The following option is supported: <i>generic_options</i> options supported by the generic volcopy command. See volcopy (1M).
OPERANDS	The following operands are supported: <i>fsname</i> represents the mount point (for example, root , u1 , etc.) of the file system being copied. <i>srcdevice</i> or <i>destdevice</i> the disk partition specified using the raw device (for example, /dev/rdisk/cld0s8 , /dev/rdisk/cld1s8 , etc.). <i>srcdevice</i> and <i>volname1</i> the device and physical volume from which the copy of the file system is being extracted. <i>destdevice</i> and <i>volname2</i> the target device and physical volume. <i>fsname</i> and <i>volname</i> are limited to six or fewer characters and recorded in the superblock. <i>volname</i> may be '-' to use the existing volume name.
EXIT STATUS	The following exit values are returned: 0 Successful file system copy. non-zero An error has occurred.
FILES	/var/adm/filesave.log a record of file systems/volumes copied
SEE ALSO	cpio (1), dd (1M), labelit (1M), volcopy (1M), fs_ufs (4)
NOTES	volcopy does not support copying to tape devices. Use dd (1M) for copying to and from tape devices.

NAME	vold – Volume Management daemon to manage CD-ROM and floppy devices														
SYNOPSIS	<code>/usr/sbin/vold [-n] [-t] [-v] [-f <i>config-file</i>] [-l <i>log-file</i>] [-d <i>root-dir</i>] [-L <i>debug-level</i>]</code>														
AVAILABILITY	SUNWvolu														
DESCRIPTION	<p>The Volume Management daemon, vold, creates and maintains a file system image rooted at <i>root-dir</i> that contains symbolic names for floppies and CD-ROMs. The default <i>root-dir</i> is set to <code>/vol</code> if no directory is specified by the <code>-d</code> option.</p> <p>vold reads the <code>/etc/vold.conf</code> configuration file upon startup. If the configuration file is modified later, vold must be told to reread the <code>/etc/vold.conf</code> file. Do this by entering</p> <p style="text-align: center;">example# kill -HUP vold_pid</p>														
OPTIONS	<p>The following options are supported:</p> <ul style="list-style-type: none"> <code>-n</code> Never writeback. Volume Management updates media labels with unique information if labels are not unique. This flag keeps Volume Management from changing your media. The default setting is FALSE. <code>-t</code> Dump NFS trace information to the log file. The default setting is FALSE. <code>-v</code> Provide lots of status information to the log file. The default setting is FALSE (do not provide status info to log file). <code>-d <i>root-dir</i></code> Specify an alternate root directory. The default location is <code>/vol</code>. Setting this will also cause other Volume Management utilities to use this as the default root directory. <code>-f <i>config-file</i></code> Specify an alternate configuration file. The default file is <code>/etc/vold.conf</code>. <code>-l <i>log-file</i></code> Specify an alternate log file. The default log file is <code>/var/adm/vold.log</code>. <code>-L <i>debug-level</i></code> Change the level (verbosity) of debug messages sent to the log file. The range is 0 to 99 where 0 is nothing and 99 is everything. The default level is 0. 														
ENVIRONMENT	<p>vold sets the following environment variables to aid programs which are called when events such as insert, notify, and eject occur:</p> <table border="0" style="margin-left: 2em;"> <tr> <td>VOLUME_ACTION</td> <td>Event that caused this program to be executed.</td> </tr> <tr> <td>VOLUME_PATH</td> <td>Pathname of the matched <i>regex</i> from the <code>vold.conf</code> file.</td> </tr> <tr> <td>VOLUME_DEVICE</td> <td>Device (in <code>/vol/dev</code>) that applies to the media.</td> </tr> <tr> <td>VOLUME_NAME</td> <td>Name of the volume in question.</td> </tr> <tr> <td>VOLUME_USER</td> <td>User ID of the user causing the event to occur.</td> </tr> <tr> <td>VOLUME_SYMNAME</td> <td>Symbolic name of a device containing the volume.</td> </tr> <tr> <td>VOLUME_MEDIATYPE</td> <td>Name of the type of media (CD-ROM or floppy)</td> </tr> </table>	VOLUME_ACTION	Event that caused this program to be executed.	VOLUME_PATH	Pathname of the matched <i>regex</i> from the <code>vold.conf</code> file.	VOLUME_DEVICE	Device (in <code>/vol/dev</code>) that applies to the media.	VOLUME_NAME	Name of the volume in question.	VOLUME_USER	User ID of the user causing the event to occur.	VOLUME_SYMNAME	Symbolic name of a device containing the volume.	VOLUME_MEDIATYPE	Name of the type of media (CD-ROM or floppy)
VOLUME_ACTION	Event that caused this program to be executed.														
VOLUME_PATH	Pathname of the matched <i>regex</i> from the <code>vold.conf</code> file.														
VOLUME_DEVICE	Device (in <code>/vol/dev</code>) that applies to the media.														
VOLUME_NAME	Name of the volume in question.														
VOLUME_USER	User ID of the user causing the event to occur.														
VOLUME_SYMNAME	Symbolic name of a device containing the volume.														
VOLUME_MEDIATYPE	Name of the type of media (CD-ROM or floppy)														

FILES	/etc/vold.conf	Volume Management daemon configuration file. Directs the Volume Management daemon to control certain devices, and causes events to occur when specific criteria are met.
	/usr/lib/vold/*.so.1	Shared objects called by Volume Management daemon when certain actions occur.
	/var/adm/vold.log	the default log file location (see the -l option for a description).
	/vol	the default Volume Management root directory.

SEE ALSO **volcancel(1)**, **volcheck(1)**, **volmissing(1)** **rmmount(1M)**, **rmmount.conf(4)**, **vold.conf(4)**, **volfs(7FS)**,
System Administration Guide, Volume I

NAME	wall – write to all users
SYNOPSIS	<code>/usr/sbin/wall [-a] [-g <i>grpname</i>] [<i>filename</i>]</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>wall reads its standard input until an end-of-file. It then sends this message to all currently logged-in users preceded by:</p> <p style="text-align: center;">Broadcast Message from ...</p> <p>If <i>filename</i> is given, then the message is read in from that file. Normally, pseudo-terminals that do not correspond to rlogin sessions are ignored. Thus, when using a window system, the message appears only on the console window. However, -a will send the message even to such pseudo-terminals.</p> <p>It is used to warn all users, typically prior to shutting down the system.</p> <p>The sender must be super-user to override any protections the users may have invoked (see mesg(1)).</p> <p>wall runs setgid() (see setuid(2)) to the group ID tty, in order to have write permissions on other user's terminals.</p> <p>wall will detect non-printable characters before sending them to the user's terminal. Control characters will appear as a " ^ " followed by the appropriate ASCII character; characters with the high-order bit set will appear in "meta" notation. For example, '\003' is displayed as '^C' and '\372' as '^M-z'.</p>
OPTIONS	<p>-a broadcast message to the console and pseudo-terminals.</p> <p>-g <i>grpname</i> broadcast to a specified group only.</p>
ENVIRONMENT	<p>If the LC_* variables (LC_CTYPE, LC_TIME, LC_COLLATE, LC_NUMERIC, and LC_MONETARY) (see environ(5)) are not set in the environment, the operational behavior of wall for each corresponding locale category is determined by the value of the LANG environment variable. If LC_ALL is set, its contents are used to override both the LANG and the other LC_* variables. If none of the above variables are set in the environment, the "C" (U.S. style) locale determines how wall behaves.</p>
FILES	<code>/dev/tty*</code>
SEE ALSO	mesg(1) , write(1) , setuid(2) , environ(5)
NOTES	“Cannot send to ...” when the open on a user's tty file fails.

NAME	whodo – who is doing what
SYNOPSIS	<code>/usr/sbin/whodo [-h] [-l] [user]</code>
AVAILABILITY	SUNWcsu
DESCRIPTION	<p>whodo produces formatted and dated output from information in the <code>/var/adm/utmp</code>, <code>/tmp/ps_data</code>, and <code>/proc/pid</code> files.</p> <p>The display is headed by the date, time, and machine name. For each user logged in, device name, user-ID and login time is shown, followed by a list of active processes associated with the user-ID. The list includes the device name, process-ID, CPU minutes and seconds used, and process name.</p> <p>If <i>user</i> is specified, output is restricted to all sessions pertaining to that user.</p>
OPTIONS	<p>-h Suppress the heading.</p> <p>-l Produce a long form of output. The fields displayed are: the user's login name, the name of the tty the user is on, the time of day the user logged in (in <i>hours:minutes</i>), the idle time — that is, the time since the user last typed anything (in <i>hours:minutes</i>), the CPU time used by all processes and their children on that terminal (in <i>minutes:seconds</i>), the CPU time used by the currently active processes (in <i>minutes:seconds</i>), and the name and arguments of the current process.</p>
EXAMPLES	<p>The command:</p> <pre>example% whodo</pre> <p>produces a display like this:</p> <pre>Tue Mar 12 15:48:03 1985 bailey tty09 mcn 8:51 tty09 28158 0:29 sh tty52 bdr 15:23 tty52 21688 0:05 sh tty52 22788 0:01 whodo tty52 22017 0:03 vi tty52 22549 0:01 sh xt162 lee 10:20 tty08 6748 0:01 layers xt162 6751 0:01 sh xt163 6761 0:05 sh tty08 6536 0:05 sh</pre>

ENVIRONMENT

If any of the `LC_*` variables (`LC_CTYPE`, `LC_MESSAGES`, `LC_TIME`, `LC_COLLATE`, `LC_NUMERIC`, and `LC_MONETARY`) (see `environ(5)`) are not set in the environment, the operational behavior of `tar(1)` for each corresponding locale category is determined by the value of the `LANG` environment variable. If `LC_ALL` is set, its contents are used to override both the `LANG` and the other `LC_*` variables. If none of the above variables is set in the environment, the "C" (U.S. style) locale determines how `tar` behaves.

LC_CTYPE

Determines how `tar` handles characters. When `LC_CTYPE` is set to a valid value, `tar` can display and handle text and filenames containing valid characters for that locale. `tar` can display and handle Extended Unix code (EUC) characters where any individual character can be 1, 2, or 3 bytes wide. `tar` can also handle EUC characters of 1, 2, or more column widths. In the "C" locale, only characters from ISO 8859-1 are valid.

LC_MESSAGES

Determines how diagnostic and informative messages are presented. This includes the language and style of the messages, and the correct form of affirmative and negative responses. In the "C" locale, the messages are presented in the default form found in the program itself (in most cases, U.S. English).

LC_TIME

Determines how `tar` handles date and time formats. In the "C" locale, date and time handling follow the U.S. rules.

FILES

`/etc/passwd`
`/tmp/ps_data`
`/var/adm/utmp`
`/proc/pid`

SEE ALSO

`ps(1)`, `tar(1)`, `who(1)`, `environ(5)`

DIAGNOSTICS

If the PROC driver is not installed or configured or if `/proc` is not mounted, a message to that effect is issued and `whodo` will fail.

The exit status is zero on success, non-zero on failure.

NAME	ypbind – NIS binder process
SYNOPSIS	<code>/usr/lib/netsvc/yp/ypbind [-broadcast -ypset -ypsetme]</code>
DESCRIPTION	<p>NIS provides a simple network lookup service consisting of databases and processes. The databases are stored at the machine that runs an NIS server process. The programmatic interface to NIS is described in ypclnt(3N). Administrative tools are described in ypinit(1M), ypwhich(1), and ypset(1M). Tools to see the contents of NIS maps are described in ypcat(1), and ypmatch(1).</p> <p>ypbind is a daemon process that is activated at system startup time from the startup script <code>/etc/init.d/rpc</code>. By default, it is invoked as ypbind -broadcast. ypbind runs on all client machines that are set up to use NIS, see sysidtool(1M).</p> <p>The function of ypbind is to remember information that lets all NIS client processes on a node communicate with some NIS server process. ypbind must run on every machine which has NIS client processes. The NIS server may or may not be running on the same node, but must be running somewhere on the network. The NIS server is not supported in SunOS releases anymore, but ypbind can communicate with an NIS server, ypserv, on an earlier SunOS release or an NIS+ server in "YP-compatibility mode", see rpc.nisd(1M). Refer to the NOTES section in ypfiles(4) for implications of being served by such an NIS+ server.</p> <p>The information ypbind remembers is called a <i>binding</i> — the association of a domain name with a NIS server.</p> <p>The process of binding is driven by client requests. As a request for an unbound domain comes in, if started with the -broadcast option, the ypbind process broadcasts on the net trying to find an NIS server, either a ypserv process serving the domain or an rpc.nisd process in "YP-compatibility mode" serving NIS+ directory with name the same as (case sensitive) the domain in the client request. Since the binding is established by broadcasting, there must be at least one NIS server on the net.</p> <p>If started without the -broadcast option, ypbind process steps through the list of NIS servers that was created by ypinit -c for the requested domain. There must be an NIS server process on at least one of the hosts in the NIS servers file. All the hosts in the NIS servers file must be listed in <code>/etc/hosts</code> along with their IP addresses. Once a domain is bound by ypbind, that same binding is given to every client process on the node. The ypbind process on the local node or a remote node may be queried for the binding of a particular domain by using the ypwhich(1) command.</p> <p>If ypbind is unable to speak to the NIS server process it is bound to, it marks the domain as unbound, tells the client process that the domain is unbound, and tries to bind the domain once again. Requests received for an unbound domain will wait until the requested domain is bound. In general, a bound domain is marked as unbound when the node running the NIS server crashes or gets overloaded. In such a case, ypbind will try to bind to another NIS server using the process described above.</p>

ypbind also accepts requests to set its binding for a particular domain. The request is usually generated by the **ypset**(1M) command. In order for **ypset** to work, **ypbind** must have been invoked with flags **-ypset** or **-ypsetme**.

- OPTIONS**
- broadcast** Send a broadcast datagram using UDP/IP that requests the information needed to bind to a specific NIS server. This option is analogous to **ypbind** with no options in earlier Sun releases and is recommended for ease of use.
 - ypset** Allow users from any remote machine to change the binding via the **ypset** command. By default, no one can change the binding. This option is really insecure.
 - ypsetme** Only allow **root** on the local machine to change the binding to a desired server via the **ypset** command. **ypbind** can verify the caller is indeed a **root** user by accepting such requests only on the loopback transport. By default, no external process can change the binding.

FILES */var/yp/binding/ypdomain/ypservers*
/etc/hosts

SEE ALSO *ypcat*(1), *ypmatch*(1), *ypwhich*(1), *ifconfig*(1M), *rpc.nisd*(1M), *ypinit*(1M), *ypset*(1M), *ypclnt*(3N), *hosts*(4), *ypfiles*(4)

NOTES **ypbind** supports multiple domains. The **ypbind** process can maintain bindings to several domains and their servers.

The **-broadcast** option works only on the UDP transport. It is insecure since it trusts "any" machine on the net that responds to the broadcast request and poses itself as an NIS server.

NAME	ypinit – set up NIS client						
SYNOPSIS	<code>/usr/sbin/ypinit [-c] [-m] [-s <i>master_server</i>]</code>						
AVAILABILITY	SUNWnisu						
DESCRIPTION	<p>ypinit can be used to set up an NIS client system. You must be the super-user to run this command. This script need not be used at all if ypbind(1M) is started with the -broadcast option (it is invoked with this option from the start up script <code>/etc/init.d/rpc</code>).</p> <p>Normally, ypinit is run only once after installing the system. It may be run whenever a new NIS server is added to the network or an existing one is decommissioned.</p> <p>ypinit prompts for a list of NIS servers to bind the client to; this list should be ordered from the closest to the furthest server. Each of these NIS servers <i>must</i> be listed in <code>/etc/hosts</code> along with its IP address. ypinit stores the list in file <code>/var/yp/binding/domain/ypservers</code>. This file is used by ypbind when run without the -broadcast option.</p>						
OPTIONS	<table><tr><td>-c</td><td>Set up a ypclient system.</td></tr><tr><td>-m</td><td>Build a master ypserver data base.</td></tr><tr><td>-s <i>master_server</i></td><td>Slave data base. <i>master_server</i> must be an existing reachable ypserver.</td></tr></table>	-c	Set up a ypclient system.	-m	Build a master ypserver data base.	-s <i>master_server</i>	Slave data base. <i>master_server</i> must be an existing reachable ypserver.
-c	Set up a ypclient system.						
-m	Build a master ypserver data base.						
-s <i>master_server</i>	Slave data base. <i>master_server</i> must be an existing reachable ypserver.						
FILES	<code>/etc/hosts</code> <code>/var/yp/binding/domain/ypservers</code>						
SEE ALSO	ypbind (1M), sysinfo (2), hosts (4)						
BUGS	ypinit sets up the list of NIS servers only for the current domain on the system when it is run, that is, the domain returned by the SI_SRPC_DOMAIN command to sysinfo (2). Care should be taken to ensure that this is the same as the desired domain for NIS client processes.						

NAME	ypmake – rebuild NIS database
DESCRIPTION	The server for NIS version 2, formerly known as YP, is not supported in SunOS anymore. It has been replaced by NIS+, the new version of the Network Information Service, see nis+(1) . This release only supports the client functionality of NIS, see ypclnt(3N) . The client functions are either supported by the ypserv process running on a machine with an earlier version of SunOS or by the NIS+ server in "YP-compatibility" mode, see rpc.nisd(1M) . See ypfiles(4) for more details.
SEE ALSO	nis+(1) , rpc.nisd(1M) , ypbind(1M) , ypclnt(3N) , ypfiles(4)

NAME	ypoll – return current version of a NIS map at a NIS server host
SYNOPSIS	<i>/usr/sbin/ypoll [-d ydomain] [-h host] mapname</i>
AVAILABILITY	SUNWnisu
DESCRIPTION	The ypoll command asks a ypserv() process what the order number is, and which host is the master NIS server for the named map.
OPTIONS	-d ydomain Use <i>ydomain</i> instead of the default domain. -h host Ask the ypserv process at <i>host</i> about the map parameters. If <i>host</i> is not specified, the NIS server for the local host is used. That is, the default host is the one returned by ypwhich(1) .
SEE ALSO	ypwhich(1) , ypfiles(4)

NAME	ypset – point ypbind at a particular server
SYNOPSIS	<code>/usr/sbin/ypset [-d <i>ypdomain</i>] [-h <i>host</i>] <i>server</i></code>
AVAILABILITY	SUNWnisu
DESCRIPTION	<p>In order to run ypset, ypbind must be initiated with the -ypset or -ypsetme options. See ypbind(1M). ypset tells ypbind to get NIS services for the specified <i>ypdomain</i> from the ypserv process running on <i>server</i>. If <i>server</i> is down, or is not running ypserv, this may not be discovered until an NIS client process tries to get a binding for the domain. At this point, the binding set by ypset will be tested by ypbind. If the binding is invalid, ypbind will attempt to rebind for the same domain.</p> <p>ypset is useful for binding a client node which is not on a broadcast net, or is on a broadcast net which is not running a NIS server host. It also is useful for debugging NIS client applications, for instance where a NIS map only exists at a single NIS server host.</p> <p>In cases where several hosts on the local net are supplying NIS services, it is possible for ypbind to rebind to another host even while you attempt to find out if the ypset operation succeeded. For example, you can type:</p> <pre style="margin-left: 40px;">example% ypset host1 example% ypwhich host2</pre> <p>which can be confusing. This is a function of the NIS subsystem's attempt to load-balance among the available NIS servers, and occurs when host1 does not respond to ypbind because it is not running ypserv (or is overloaded), and host2, running ypserv, gets the binding.</p> <p><i>server</i> indicates the NIS server to bind to, and must be specified as a name or an IP address. This will work only if the node has a current valid binding for the domain in question, and ypbind has been set to allow use of ypset. In most cases, <i>server</i> should be specified as an IP address.</p> <p>ypset tries to bind over a connectionless transport. The NIS library call, yp_all(), uses connection-oriented transport and derives the NIS server's address based on the connectionless address supplied by ypset.</p> <p>Refer to ypfiles(4) for an overview of the NIS name service.</p>
OPTIONS	<p>-d <i>ypdomain</i> Use <i>ypdomain</i>, instead of the default domain.</p> <p>-h <i>host</i> Set ypbind's binding on <i>host</i>, instead of locally. <i>host</i> must be specified as a name.</p>
SEE ALSO	ypwhich(1) , ypfiles(4)

NAME	ypxfr – transfer NIS map from a NIS server to host
SYNOPSIS	<code>/usr/lib/netsvc/yp/ypxfr [-c] [-f] [-C tid prog server] [-d ydomain] [-h host] [-s ydomain] mapname</code>
DESCRIPTION	<p>The ypxfr command moves a NIS map in the default domain for the local host to the local host by making use of normal NIS services. It creates a temporary map in the directory <code>/var/yp/ydomain</code> (this directory must already exist; <i>ydomain</i> is the default domain for the local host), fills it by enumerating the map's entries, fetches the map parameters (master and order number), and loads them. It then deletes any old versions of the map and moves the temporary map to the real <i>mapname</i>.</p> <p>If run interactively, ypxfr writes its output to the terminal. However, if it is started without a controlling terminal, and if the log file <code>/var/yp/ypxfr.log</code> exists, it appends all its output to that file. Since ypxfr is most often run from the privileged user's crontab file, or by ypserv, the log file can be used to retain a record of what was attempted, and what the results were.</p> <p>For consistency between servers, ypxfr should be run periodically for every map in the NIS data base. Different maps change at different rates: a map may not change for months at a time, for instance, and may therefore be checked only once a day. Some maps may change several times per day. In such a case, you may want to check hourly for updates. A crontab(1) entry can be used to perform periodic updates automatically. Rather than having a separate crontab entry for each map, you can group commands to update several maps in a shell script. Examples (mnemonically named) are in <code>/usr/sbin/yp: ypxfr_1perday</code>, and <code>ypxfr_1perhour</code>. They can serve as reasonable first cuts.</p> <p>Refer to ypfiles(4) for an overview of the NIS name service.</p>
OPTIONS	<p>-c Do not send a "Clear current map" request to the local ypserv process. Use this flag if ypserv is not running locally at the time you are running ypxfr. Otherwise, ypxfr complains that it cannot talk to the local ypserv, and the transfer fails.</p> <p>-f Force the transfer to occur even if the version at the master is not more recent than the local version.</p> <p>-C tid prog server This option is <i>only</i> for use by ypserv. When ypserv starts ypxfr, it specifies that ypxfr should call back a yppush process at the host <i>server</i>, registered as program number <i>prog</i>, and waiting for a response to transaction <i>tid</i>.</p> <p>-d ydomain Specify a domain other than the default domain.</p> <p>-h host Get the map from <i>host</i>, regardless of what the map says the master is. If <i>host</i> is not specified, ypxfr asks the NIS service for the name of the master, and try to get the map from there. <i>host</i> must be a name.</p>

-s ydomain Specify a source domain from which to transfer a map that should be the same across domains.

FILES

/var/yp/ypxfr.log log file
/usr/sbin/yp/ypxfr_1perday script to run one transfer per day, for use with **cron(1M)**
/usr/sbin/yp/ypxfr_1perhour script for hourly transfers of volatile maps
/var/yp/ydomain NIS domain
/usr/spool/cron/crontabs/root privileged user's **crontab** file

SEE ALSO

crontab(1), **cron(1M)**, **ypfiles(4)**

NAME	zdump – time zone dumper
SYNOPSIS	zdump [-v] [-c <i>cutoffyear</i>] [<i>zonename</i> ...]
AVAILABILITY	SUNWcsu
DESCRIPTION	The zdump command prints the current time in each <i>zonename</i> named on the command line.
OPTIONS	-v For each <i>zonename</i> on the command line, print the current time, the time at the lowest possible time value, the time one day after the lowest possible time value, the times both one second before and exactly at each time at which the rules for computing local time change, the time at the highest possible time value, and the time at one day less than the highest possible time value. Each line ends with isdst=1 if the given time is Daylight Saving Time or isdst=0 otherwise. -c <i>cutoffyear</i> Cut off the verbose output near the start of the year <i>cutoffyear</i> .
FILES	/usr/share/lib/zoneinfo standard zone information directory
SEE ALSO	zic(1M) , ctime(3C)

NAME	zic – time zone compiler																				
SYNOPSIS	zic [-v] [-d <i>directory</i>] [-l <i>localtime</i>] [<i>filename ...</i>]																				
AVAILABILITY	SUNWcsu																				
DESCRIPTION	<p>zic reads text from the file(s) named on the command line and creates the time conversion information files specified in this input. If a <i>filename</i> is '-', the standard input is read.</p> <p>Input lines are made up of fields. Fields are separated by any number of white space characters. Leading and trailing white space on input lines is ignored. A number-sign (#) indicates a comment and extends to the end of the line. White space characters and number-signs may be enclosed within double quotes (" ") if they are to be used as part of a field. Any line that is blank (after comment stripping) is ignored. Non-blank lines are expected to be of one of three types: rule lines, zone lines, or link lines.</p>																				
Rule	<p>A rule line has the form:</p> <table border="0"> <tr> <td>Rule</td> <td>NAME</td> <td>FROM</td> <td>TO</td> <td>TYPE</td> <td>IN</td> <td>ON</td> <td>AT</td> <td>SAVE</td> <td>LETTER/S</td> </tr> </table> <p>For example:</p> <table border="0"> <tr> <td>Rule</td> <td>USA</td> <td>1969</td> <td>1973</td> <td>-</td> <td>Apr</td> <td>lastSun</td> <td>2:00</td> <td>1:00</td> <td>D</td> </tr> </table> <p>The fields that make up a rule line are:</p> <p>NAME Gives the (arbitrary) name of the set of rules this rule is part of.</p> <p>FROM Gives the first year in which the rule applies. The word minimum (or an abbreviation) means the minimum year with a representable time value. The word maximum (or an abbreviation) means the maximum year with a representable time value.</p> <p>TO Gives the final year in which the rule applies. In addition to minimum and maximum (as above), the word only (or an abbreviation) may be used to repeat the value of the FROM field.</p> <p>TYPE Gives the type of year in which the rule applies. If TYPE is '-' then the rule applies in all years between FROM and TO inclusive; if TYPE is uspres, the rule applies in U.S. Presidential election years; if TYPE is nonpres, the rule applies in years other than U.S. Presidential election years. If TYPE is something else, then zic executes the command</p> <p style="text-align: center;">yearistype year type</p> <p>to check the type of a year: an exit status of 0 means that the year is of the given type; an exit status of 1 means that the year is not of the given type.</p> <p>IN Names the month in which the rule takes effect. Month names may be abbreviated.</p>	Rule	NAME	FROM	TO	TYPE	IN	ON	AT	SAVE	LETTER/S	Rule	USA	1969	1973	-	Apr	lastSun	2:00	1:00	D
Rule	NAME	FROM	TO	TYPE	IN	ON	AT	SAVE	LETTER/S												
Rule	USA	1969	1973	-	Apr	lastSun	2:00	1:00	D												

ON Gives the day on which the rule takes effect. Recognized forms include:

5 the fifth of the month

lastSun the last Sunday in the month

lastMon the last Monday in the month

Sun>=8 first Sunday on or after the eighth

Sun<=25 last Sunday on or before the 25th

Names of days of the week may be abbreviated or spelled out in full.
Note: There can not be spaces within the ON field.

AT Gives the time of day at which the rule takes effect. Recognized forms include:

2 time in hours

2:00 time in hours and minutes

15:00 24-hour format time (for times after noon)

1:28:14 time in hours, minutes, and seconds

Any of these forms may be followed by the letter **w** if the given time is local “wall clock” time or **s** if the given time is local “standard” time; in the absence of **w** or **s**, wall clock time is assumed.

SAVE Gives the amount of time to be added to local standard time when the rule is in effect. This field has the same format as the AT field (without the **w** and **s** suffixes).

LETTER/S Gives the “variable part” (for example, the “S” or “D” in “EST” or “EDT” of time zone abbreviations to be used when this rule is in effect. If this field is ‘-’, the variable part is null.

Zone

A zone line has the form:

Zone **NAME** **GMTOFF** **RULES/SAVE** **FORMAT** **[UNTIL]**

For example:

Zone **Australia/South-west** **9:30** **CST** **1992 Mar 15** **12:00**

The fields that make up a zone line are:

NAME The name of the time zone. This is the name used in creating the time conversion information file for the zone.

GMTOFF The amount of time to add to GMT to get standard time in this zone. This field has the same format as the AT and SAVE fields of rule lines; begin the field with a minus sign if time must be subtracted from GMT.

RULES/SAVE The name of the rule(s) that apply in the time zone or, alternately, an amount of time to add to local standard time. If this field is ‘-’ then standard time always applies in the time zone.

FORMAT The format for time zone abbreviations in this time zone. The pair of characters %s is used to show where the “variable part” of the time zone

abbreviation goes.

UNTIL The time at which the GMT offset or the rule(s) change for a location. It is specified as a year, a month, a day, and a time of day. If this is specified, the time zone information is generated from the given GMT offset and rule change until the time specified.

The next line must be a “continuation” line; this has the same form as a zone line except that the string “Zone” and the name are omitted, as the continuation line will place information starting at the time specified as the **UNTIL** field in the previous line in the file used by the previous line. Continuation lines may contain an **UNTIL** field, just as zone lines do, indicating that the next line is a further continuation.

Link A link line has the form:

Link **LINK-FROM** **LINK-TO**

For example:

Link **US/Eastern** **EST5EDT**

The **LINK-FROM** field should appear as the **NAME** field in some zone line; the **LINK-TO** field is used as an alternate name for that zone.

Except for continuation lines, lines may appear in any order in the input.

OPTIONS

- v** Complain if a year that appears in a data file is outside the range of years representable by system time values (0:00:00 a.m. GMT, January 1, 1970, to 3:14:07 a.m. GMT, January 19, 2038).
- d *directory*** Create time conversion information files in the directory *directory* rather than in the standard directory **/usr/share/lib/zoneinfo**.
- l *timezone*** Use the time zone *timezone* as local time. **zic** will act as if the file contained a link line of the form:

Link *timezone* **localtime**

FILES **/usr/share/lib/zoneinfo** standard directory used for created files

SEE ALSO **time(1)**, **ctime(3C)**

NOTES For areas with more than two types of local time, you may need to use local standard time in the **AT** field of the earliest transition time’s rule to ensure that the earliest transition time recorded in the compiled file is correct.

Index

Special Characters

/devices directory

configure — drvconfig, 1M-169

2

24-bit TrueColor Visual for OpenWindows with an S24 frame buffer system (TCX) — tcxconfig
configure the default linearity, 1

A

accept — accept print requests, 1M-20

accounting

run daily — runacct, 1M-550

accounting commands

- acctcms, 1M-24
- acctcon, 1M-26
- acctdisk, 1M-21
- acctdusg, 1M-21
- acctmerg, 1M-28
- accton, 1M-21
- acctprc, 1M-29
- acctwtmp, 1M-21
- closewtmp, 1M-21
- fwtmp, 1M-228
- utmp2wtmp, 1M-21
- wtmpfix, 1M-228

accounting shell procedures

— chargefee, 1M-31

accounting shell procedures, *continued*

- ckpacct, 1M-31
- dodisk, 1M-31
- lastlogin, 1M-31
- monacct, 1M-31
- nulladm, 1M-31
- prctmp, 1M-31
- prdaily, 1M-31
- prtacct, 1M-31
- runacct, 1M-31
- shutacct, 1M-31
- startup, 1M-31
- turnacct, 1M-31

acct — miscellaneous accounting commands, 1M-21

acctcms — command summary from process accounting records, 1M-24

acctcon — create summary of connect records, 1M-26

acctcon1 — create summary of connect records, 1M-26

acctcon2 — create summary of connect records, 1M-26

acctdisk — create disk usage records, 1M-21

acctdusg — compute disk usage by login, 1M-21

acctmerg — merge or add total disk accounting files, 1M-28

accton — turn on process accounting, 1M-21

acctprc — summary of process accounting, 1M-29
acctprc1 — summary of process accounting, 1M-29
acctprc2 — summary of process accounting, 1M-29
acctsh — shell procedures for accounting, 1M-31
acctwtmp — create connect accounting record in /var/adm/wtmp, 1M-21
adb scripts
 generate — adbgen, 1M-34
adbgen — generate adb script, 1M-34
add a new device driver to the system — add_drv, 1M-36
add_drv — add a new device driver to the system, 1M-36
add_install_client — scripts used to install the Solaris software, 1M-296
addbadsec — map out defective disk blocks, 1M-38
address resolution display and control — arp, 1M-43
administer disk space used for caching file systems with the Cache File-System (CacheFS) — cfsadmin, 1M-103
administer NIS+ aliases
 — aliasadm, 1M-41
admintool — system administration tool with a graphical interface, 1M-39
aliasadm — administer NIS+ aliases, 1M-41
allocate — allocate devices, 1M-42
analyze the SunOS 4.x client support after converting to Solaris 2.x — discover4x, 1M-159
arp — address resolution display and control, 1M-43
ASET
 monitors or restricts accesses to system files and directories — aset, 1M-45
 prints tasks status — taskstat, 1M-661
aset.restore — restore files and directories to the state before ASET was installed, 1M-50
aspppd — Asynchronous PPP Link Manager, 1M-51

aspppls — Asynchronous PPP Link Manager, 1M-51
Asynchronous PPP Link Manager — aspppd, 1M-51
audit — maintain audit trail, 1M-59
audit records
 select or merge from audit trail files — auditreduce, 1M-68
audit statistics report — auditstat, 1M-74
audit trail file
 select records from — auditreduce, 1M-68
audit_startup shell script, 1M-60
audit_warn — audit daemon warning script, 1M-61
auditconfig — get and set kernel audit parameters, 1M-63
auditd — audit daemon, 1M-66
auditreduce — select or merge audit records from audit trail files, 1M-68
auditstat — display kernel audit statistics, 1M-74
autoboot procedures — boot, 1M-299
Autofs
 automatically mount file systems — automount, 1M-76
 mount/unmount request server — automountd, 1M-82
Automated Security Enhancement Tool, See ASET
automount — automatically mount file systems, 1M-76
automountd — Autofs mount/unmount request server, 1M-82
autopush — configures lists of automatically pushed STREAMS modules, 1M-83

B

Basic Security Module
 enable — bsmconv, bsmunconv, 1M-96
Basic Security Module commands
 — audit, 1M-59
 — audit_startup, 1M-60
 — audit_warn, 1M-61
 — auditconfig, 1M-63
 — auditd, 1M-66

Basic Security Module commands, *continued*

- auditreduce, 1M-68
- auditstat, 1M-74
- bdconfig — configures the bd (buttons and dials) stream, 1M-85
- boot — bootstrap procedures, 1M-87
- boot — system startup procedures, 1M-299
- boot parameter server — rpc.bootparamd, 1M-525
- bootstrap procedures — boot, 1M-299
- bootstrap PROM monitor program — monitor, 1M-373
- Bourne shell
 - rsh, 1M-547
- broadcast message
 - network wall server — rpc.rwall, 1M-535
 - write to all users — wall, 1M-706
 - write to all users of a group — wall, 1M-706
 - write to all users over a network — rwall, 1M-552
- bsmconv — enable Solaris system to use Basic Security Module, 1M-96
- bsmunconv — enable Solaris system to use Basic Security Module, 1M-96

C

C compiler

- create a front-end to cc command — gcc, 1M-229

cacheFS

- cfsadmin — administer disk space used for caching file systems with the Cache File-System (CacheFS), 1M-103

cachefs

- fsck_cachefs — check integrity of data cached with CacheFS, 1M-210
- mount_cachefs — mount CacheFS file systems, 1M-389

cachefslog — Cache File System logging, 1M-97

cachefsstat — Cache File System statistics, 1M-98

cachefswssize — Determine working set size for cachefs, 1M-99

captain — convert termcap description to terminfo description, 1M-100

catman — create cat files for manual pages, 1M-101

CD-ROM

packages on — pkgadd, 1M-478

CD-ROM, managing

— rmmount, 1M-519

CD-ROM, mounting

— rmmount, 1M-519

cfs

fsck_cachefs — check integrity of data cached with CacheFS, 1M-210

mount_cachefs — mount CacheFS file systems, 1M-389

cfsadmin — administer disk space used for caching file systems with the Cache File-System (CacheFS), 1M-103

CacheFS Resource Parameters, 1M-104

cg14config — configure the cgfourteen device, 1M-106

cg14config — configure the cgfourteen device, 1M-106

character classification

generate table — chrtbl, 1M-110

character conversion

generate character conversion tables — wchrtbl, 1M-110

chargefee — accounting shell procedure, 1M-31

check — scripts used to install the Solaris software, 1M-296

check integrity of data cached with CacheFS —

fsck_cachefs, 1M-210

chown — change owner of file, 1M-108

chroot — change root directory for a command, 1M-109

chrtbl — generate character classification tables, 1M-110

ckpacct — accounting shell procedure, 1M-31

clear inode

— clri, 1M-115

— dcopy, 1M-115

closewtmp — puts a false DEAD_PROCESS record in /var/adm/wtmp file, 1M-21

clri — clear inode, 1M-115
 colltbl — create collation database, 1M-116
 configuration applications
 execute or define — sysidconfig, 1M-655
 configure
 cgfourteen device — cg14config, 1M-106
 contiguous memory for sx video — sxcon-
 fig, 1M-641
 undo system configuration — sys_unconfig,
 1M-651
 configure device attributes — devconfig, 1M-140
 configures the bd (buttons and dials) stream —
 bdconfig, 1M-85
 connect accounting
 — acctcon, 1M-26
 — fwtmp, 1M-228
 — utmp2wtmp, 1M-21
 — wtmpfix, 1M-228
 convert4x — converts SunOS 4.x clients to work
 from Solaris 2.x servers, 1M-120
 CPU PROM monitor
 program — monitor, 1M-373
 crash — examine system memory image
 create
 NIS dbm file — makedbm, 1M-359
 create FNS file system contexts — fncreate_fs,
 1M-197
 create new printers in the FNS namespace —
 fncreate_printer, 1M-201
 create or modify fixed disk partition table —
 fdisk, 1M-181
 cron — clock daemon, 1M-131

D

daemon, distributed system administration — sad-
 mind, 1M-559
 daemons
 clock daemon — cron, 1M-131
 Internet Trivial File Transfer Protocol —
 in.tftpd, 1M-281
 kernel statistics — rpc.rstatd, 1M-533
 network router discovery daemon —
 in.rdisc, 1M-265

daemons, *continued*

 network status monitor — statd, 1M-625
 network username server — rpc.rusersd,
 1M-534
 network wall server — rpc.rwall, 1M-535
 NFS — nfsd, 1M-417
 NIS+ service — rpc.nisd, 1M-526
 remote quota — rquotad, 1M-546
 remote shell server — in.rshd, 1M-274
 RPC-based remote execution — rpc.rexd,
 1M-532
 server which records packets sent by spray —
 rpc.sprayd, 1M-536

date
 set system date from a remote host — rdate,
 1M-512

dcopy — clear inode, 1M-115
 dd — convert and copy a file, 1M-133
 deallocate — deallocate devices, 1M-138
 debug tools
 generate adb script — adbgen, 1M-34
 devattr — displays the values for a device's attri-
 butes, 1M-139
 devconfig — configure device attributes, 1M-140
 device access
 verifies accessibility — getvol, 1M-238
 device_maps
 display entries — dminfo, 1M-166
 devices
 add, modify remove device entry from table —
 putdev, 1M-501
 add, remove and change device group table —
 putdgrp, 1M-504
 adds /dev entries for miscellaneous devices
 and pseudo-devices — devlinks,
 1M-144
 adds /dev entries for tape drives attached to
 the system — tapes, 1M-660
 allocation — allocate, 1M-42
 cgfourteen — configure the cgfourteen dev-
 ice, 1M-106
 deallocation — deallocate, 1M-138
 display access control entries from
 device_maps, 1M-166

devices, *continued*

display attributes — devattr, 1M-139
group list — getdgrp, 1M-232
list — getdev, 1M-230
list_devices — list_devices, 1M-322
lists members of a group — listdgrp, 1M-323
name — devnm, 1M-147
print information about disk devices — devinfo, 1M-143
print VTOC of a block device — prtvtoc, 1M-497
release from exclusive use — devfree, 1M-142
remove a device driver from the system — rm_drv, 1M-516
reserve for exclusive use — devfree, 1M-148
df — display amount of used and available disk space, 1M-150
df_ufs — display amount of space occupied by ufs, 1M-153
dfmounts — displays information on resources shared through DFS, 1M-154
dfmounts_nfs — displays information on resources shared through NFS, 1M-156
DFS
display information on resources shared — dfmounts, 1M-154
list available resources from remote or local systems — dfshares, 1M-157
dfshares — list available resources from remote or local systems, 1M-157
dfshares_nfs — list available resources from remote systems, 1M-158
Diffie-Hellman key pair, 1M-416
directories
move — mvdir, 1M-406
discover4x — analyze the SunOS 4.x client support after converting to Solaris 2.x, 1M-159
disk accounting
— acctdisk, 1M-21
— acctdusg, 1M-21
— acctmerg, 1M-28
disk blocks

disk blocks, *continued*

display free ones — df, 1M-150
disk quotas
remote quota daemon — rquotad, 1M-546
disk quotas and usage
display for a user's ufs file system — quota, 1M-509
disk usage
summary — du, 1M-171
diskettes
packages on — pkgadd, 1M-478
disks — adds /dev entries for hard disks attached to the system, 1M-160
access profiler — fusage, 1M-226
partitioning and maintenance utility — format, 1M-204
diskscan — perform surface analysis, 1M-162
display
system configuration information — prtconf, 1M-491
system diagnostic information — prtdiag, 1M-494
Distributed File System, See DFS
dmesg — create error log from system diagnostic messages, 1M-165
dminfo — display device_maps entries, 1M-166
DoD Internet
host table — gettable, 1M-235
DoD Internet format host table, convert — htable, 1M-244
dodisk — accounting shell procedure, 1M-31
domainname — display name of current domain, 1M-168
driver statistics from a synchronous serial link, report — syncstat, 1M-649
drvconfig — configure /devices, 1M-169
du — summarize disk usage, 1M-171

E

edquota — edit user quotas for ufs file system, 1M-173
EEPROM display and load program — eeprom, 1M-175

encryption keys
 server for storing private keys — `keyserv`,
 1M-313

environment variables
 convert termcap description into terminfo
 description — `captoinfo`, 1M-100

error log
 create from system diagnostic messages —
 `dmesg`, 1M-165

F

`fdetach` — detach a name from STREAMS-based
 file descriptor, 1M-180

`fdisk` — create or modify fixed disk partition table,
 1M-181
 Menu Options, 1M-181

FIFO
 make FIFO special file — `mkfifo`, 1M-363

FIFO, make — `mknod`, 1M-369

file system
 administer disk space used for caching file sys-
 tems with the Cache File-System
 (CacheFS) — `cfsadmin`, 1M-103
 change the dynamic parameters — `tunefs`,
 1M-670
 check and repair — `fsck`, 1M-207
 check and repair ufs — `fsck_ufs`, 1M-213
 construct — `mkfs`, 1M-365
 construct ufs — `mkfs_ufs`, 1M-366
 create new — `newfs`, 1M-413
 debugger — `fsdb`, 1M-216
 disk access profiler — `fusage`, 1M-226
 display amount of used disk space occupied by
 ufs — `df_ufs`, 1M-153
 display amount of used and available disk
 space — `df`, 1M-150
 link and unlink files and directories — `link`,
 1M-321
 list file names and statistics — `ff`, 1M-186
 list file names and statistics for ufs — `ff_ufs`,
 1M-188
 loopback — `mount`, 1M-386
 make literal copy — `volcopy`, 1M-701
 mount — `mount`, 1M-386, 1M-403

file system, *continued*
 mount ufs — `mount_ufs`, 1M-401
 move directory — `mmdir`, 1M-406
 provide labels — `labelit`, 1M-315
 provide labels for ufs — `labelit_ufs`,
 1M-318
 report processes using file or file structure —
 `fuser`, 1M-227
 share multiple resources — `shareall`, 1M-604
 ufs, make image copy — `volcopy_ufs`,
 1M-703
 unmount — `umount`, 1M-386, 1M-403
 unshare multiple resources — `unshareall`,
 1M-604

file system consistency check and interactive repair
 — `fsck_s5fs`, 1M-211

file system type, determine — `fstyp`, 1M-225

File Transfer Protocol
 server — `in.ftpd`, 1M-254

files
 change ownership — `chown`, 1M-108
 convert and copy — `dd`, 1M-133

firmware — bootable firmware programs and
 firmware commands, 1M-189

floppy, managing
 — `rmmount`, 1M-519

floppy, mounting
 — `rmmount`, 1M-519

flush disk activity — `sync`, 1M-643

`fmthard` — populate Volume Table of Contents on
 hard disks, 1M-190

`fncreate_fs` — create FNS file system contexts,
 1M-197

`fncreate_printer` — create new printers in the
 FNS namespace, 1M-201

`format` — disk partitioning and maintenance util-
 ity, 1M-204

`fsck` — check and repair file systems, 1M-207

`fsck_cacheufs` — check integrity of data cached
 with CacheFS, 1M-210

`fsck_s5fs` — file system consistency check and
 interactive repair, 1M-211

`fsck_ufs` — check and repair ufs, 1M-213

`fsdb` — file system debugger, 1M-216
`fsdb_ufs` — ufs file system debugger, 1M-217
 Commands, 1M-219
 Expressions, 1M-218
 Formatted Output, 1M-222
 Inode Commands, 1M-221
`fsirand` — install random inode generation numbers, 1M-224
`fstyp` — determine file system type, 1M-225
FTP
 daemon on remote host — `in.ftpd`, 1M-254
`fusage` — disk access profiler, 1M-226
`fuser` — identify processes using file or file structure, 1M-227
`fwtmp` — convert connect accounting records to ASCII, 1M-228

G

`gcc` — create a front-end to `cc` command, 1M-229
`getdev` — lists devices that match given criteria, 1M-230
`getdgrp` — lists device groups which contain devices that match given criteria, 1M-232
`getent` — gets a list of entries from administrative databases, 1M-234
`gettable` — get DoD Internet host table, 1M-235
`getty` — set terminal type, modes, speed, and line discipline, 1M-236
`getvol` — verifies device accessibility, 1M-238
group file
 check for inconsistencies — `grpck`, 1M-506
`groupadd` — add a new group definition on the system, 1M-239
`groupdel` — delete a group definition on the system, 1M-240
`groupmod` — modify a group definition on the system, 1M-241

H

`halt` — stop the processor, 1M-242
hard disks
 populate Volume Table of Contents — `fmthard`, 1M-190
`hostconfig` — configure a system's host parameters, 1M-243
hsfs
 labels for hsfs file systems, provide and print — `labelit_hsfs`, 1M-317
 mount — `mount_hsfs`, 1M-392
`htable` — convert DoD Internet format host table, 1M-244

I

I/O statistics report — `iostat`, 1M-303
ICMP
 router discovery daemon — `in.rdisc`, 1M-265
`id` — return user identity, 1M-245
`ifconfig` — configure network interface parameters, 1M-248
`in.comsat` — biff server, 1M-252
`in.fingerd` — remote user information server, 1M-253
`in.ftpd` — File Transfer Protocol daemon on remote host, 1M-254
`in.named` — Internet domain name server, 1M-261
`in.rarpd` — Reverse Address Resolution Protocol server, 1M-264
`in.rdisc` — ICMP router discovery daemon, 1M-265
`in.rexecd` — remote execution server, 1M-267
`in.rlogind` — remote login server, 1M-269
`in.routed` — network routing daemon, 1M-271
`in.rwhod` — system status server, 1M-276
`in.talkd` — talk server, 1M-278
`in.telnetd` — DARPA TELNET virtual terminal protocol server, 1M-279
`in.tftpd` — Internet Trivial File Transfer Protocol server, 1M-281
`in.tnamed` — DARPA trivial name server, 1M-282

in.uucpd — UUCP server, 1M-283
 incremental file system dump — `ufsdump`, 1M-673
 inetd — Internet services daemon, 1M-284
 init — process control initialization
 /etc/defaults/init file, 1M-290
 init and System Booting, 1M-289
 inittab Additions, 1M-290
 Run Level Changes, 1M-290
 Run Level Defined, 1M-289
 telinit, 1M-290
 init — process control initialization, 1M-289
 inode
 clear — `clri`, 1M-115
 install — install commands, 1M-120, 1M-159,
 1M-293, 1M-295
 SunOS/SVR4 — `suninstall`, 1M-636
 install_scripts — scripts used to install the
 Solaris software, 1M-296
 installboot procedures — `boot`, 1M-299
 installf — add a file to software installation
 database, 1M-300
 Internet
 biff server — `in.comsat`, 1M-252
 DARPA TELNET virtual terminal protocol
 server — `in.telnetd`, 1M-279
 DARPA trivial name server — `in.tnamed`,
 1M-282
 domain name server — `in.named`, 1M-261
 File Transfer Protocol daemon on remote host
 — `in.ftpd`, 1M-254
 ICMP router discovery daemon — `in.rdisc`,
 1M-265
 network routing daemon — `in.routed`,
 1M-271
 query domain name servers — `nslookup`,
 1M-459, 1M-463
 RARP server — `in.rarpd`, 1M-264
 remote user information server —
 `in.fingerd`, 1M-253
 send ICMP ECHO_REQUEST packets to net-
 work hosts — `ping`, 1M-476
 server for talk program — `in.talkd`, 1M-278
 services daemon — `inetd`, 1M-284
 system status server — `in.rwhod`, 1M-276

Internet, *continued*
 Trivial File Transfer Protocol server —
 `in.tftpd`, 1M-281
 Internet Control Message Protocol, See ICMP
 Internet Protocol
 to Ethernet addresses — `arp`
 Interprocess Communication
 make FIFO special file — `mkfifo`, 1M-363
 iostat — report I/O statistics, 1M-303

K

kadb — a kernel debugger, 1M-305
 kdmconfig — configure or unconfigure keyboard,
 display, and mouse options, 1M-308
 Kerberos authentication tickets
 generate and validate for kernel RPC —
 `kerbd`, 1M-310
 kernel — UNIX system executable file containing
 basic operating system services, 1M-311
 debugger — `kadb`, 1M-305
 display status of loaded modules — `modload`,
 1M-370
 load a module — `modload`, 1M-371
 SPARC Only, 1M-311, 1M-312
 unload a module — `modunload`, 1M-372
 kernel debugger — `kadb`, 1M-305
 Kernel Symbolic Debugging for OpenBoot
 Firmware — `obpsym`, 1M-466
 keyserv — server for storing private encryption
 keys, 1M-313
 killall — kill all active processes, 1M-314

L

labelit — provide labels for file systems, 1M-315
 labelit_hsf s — provide and print labels for hsf s
 file systems, 1M-317
 labelit_ufs — provide labels for ufs, 1M-318
 lastlogin — accounting shell procedure, 1M-31
 Leo (both ZX and TZ X) Graphics Accelerator —
 `leoconfig`, 1M-319
 leoconfig — initialize the Leo (both ZX and TZ X)
 Graphics Accelerator and download microcode,
 1M-319

link — link and unlink files and directories, 1M-321
 list_devices — list_devices, 1M-322
 listdgrp — lists members of a device group, 1M-323
 listen — network listener daemon, 1M-324
 lockd — network lock daemon, 1M-326
 lockfs — lock file system, 1M-327
 log files and system log daemon — syslogd, 1M-659
 logins — list user and system login information, 1M-329
 loopback file system
 mount — mount, 1M-386
 LP print services
 administer filters — lpfilter, 1M-341
 administer forms — lpforms, 1M-345
 configure — lpadmin, 1M-330
 moves queued print requests — lpmove, 1M-352
 register remote systems — lpsystem, 1M-353
 set printing queue priorities — lpusers, 1M-356
 start — lpsched, 1M-352
 stop — lpshut, 1M-352
 lpadmin — configure LP print service, 1M-330
 lpfilter — administer filters used with LP print service, 1M-341
 lpforms — administer forms used with LP print service, 1M-345
 Adding or Changing a Form, 1M-345
 Allowing and Denying Access to a Form, 1M-348
 Deleting a Form, 1M-348
 Listing Form Attributes, 1M-348
 Listing the Current Alert, 1M-350
 Removing an Alert Definition, 1M-351
 Setting an Alert to Mount a Form, 1M-349
 Terminating an Active Alert, 1M-350
 lpmove — moves print requests that are queued, 1M-352
 lpsched — start LP print service, 1M-352
 lpshut — shut LP print service, 1M-352

lpsystem — register remote systems with LP print service, 1M-353
 lpusers — set printing queue priorities, 1M-356

M

mail delivery server — sendmail, 1M-585
 mail utilities
 biff server — in.comsat, 1M-252
 mail.local — store mail in a mailbox, 1M-358
 make
 FIFO — mknod, 1M-369
 named pipe — mknod, 1M-369
 special file — mknod, 1M-369
 make FIFO special file — mkfifo, 1M-363
 make named pipe — mkfifo, 1M-363
 makedbm — make NIS ndbm file, 1M-359
 manual pages
 create cat files for — catman, 1M-101
 map out defective disk blocks — addbadsec, 1M-38
 mk — remake the binary system and commands from source code, 1M-360
 mkfifo — make FIFO special file, 1M-363
 mkfile — create a file, 1M-364
 mkfs — make file system, 1M-365
 mkfs_ufs — construct ufs, 1M-366
 mknod — make special file, 1M-369
 modinfo — display status of loaded kernel modules, 1M-370
 modload — load a kernel module, 1M-371
 modunload — unload a kernel module, 1M-372
 monacct — accounting shell procedure, 1M-31
 monetary database
 create — montbl, 1M-384
 monitor program — monitor, 1M-373
 montbl — create monetary database, 1M-384
 mount
 establish table — setmnt, 1M-597
 show all remote mounts — showmount, 1M-605
 mount — mount filesystems and remote resources, 1M-386

mount *hsfs* file systems — *mount_hfsfs*, 1M-392
mount *pcfs* file systems — *mount_pcfs*, 1M-398
mount *s5* file systems — *mount_s5fs*, 1M-399
mount_cachefs — mount CacheFS file systems,
1M-389
mount_hfsfs — mount *hsfs* file systems, 1M-392
mount_nfs — mount remote NFS resources,
1M-394
mount_pcfs — mount *pcfs* file systems, 1M-398
mount_s5fs — mount *s5* file systems, 1M-399
mount_tmpfs — mount *tmpfs*, 1M-400
mount_ufs — mount *ufs*, 1M-401
mountall — mount multiple filesystems, 1M-403
mountd — NFS mount request server, 1M-404
mouse, keyboard, display
 configure or unconfigure — *kdmconfig*,
 1M-308
mpstat — report per-processor statistics, 1M-405
mvdir — move a directory, 1M-406

N

name service cache daemon — *nscd*, 1M-457
named pipe, make — *mkfifo*, 1M-363, 1M-369
ncheck — generate a list of path names vs i-
 numbers of all files on *special*, 1M-407
ncheck_ufs — generate a list of pathnames vs i-
 numbers for *ufs*, 1M-408
ndd — get and set driver configuration parameters,
 1M-409
netstat — display network status, 1M-410
 Active Sockets (First Form), 1M-411
 Multicast Routing Tables (Fourth Form),
 1M-412
 Network Data Structures (Second Form),
 1M-411
 Routing Table (Third Form), 1M-411
 TCP Sockets, 1M-411
network
 listener daemon — *listen*, 1M-324
 lock daemon — *lockd*, 1M-326
 test connections to hosts using ICMP — *ping*,
 1M-476
Network File System, See NFS

Network Information Service, see NIS
network interface parameters
 configure — *ifconfig*, 1M-248
network listener service
 administration — *nlsadmin*
network packets capture and inspection — *snoop*,
 1M-610
network routing daemon — *in.routed*, 1M-271
network status, display — *netstat*, 1M-410
newfs — make new file system, 1M-413
newkey — create a new key in publickey database,
 1M-416
NFS
 crash and recovery functions for locking ser-
 vices — *statd*, 1M-625
 create files for use as NFS mounted swap areas
 — *mkfile*, 1M-364
 daemon — *nfsd*, 1M-417
 display information on resources shared —
 dfmounts_nfs, 1M-156
 display statistics — *nfsstat*, 1M-418
 list available resources from remote systems —
 dfshares_nfs, 1M-158
 make local filesystem available for mounting by
 remote systems — *share_nfs*,
 1M-601
 make local NFS filesystem unavailable for
 mounting by remote systems —
 unshare_nfs, 1M-685
 mount — *mount_nfs*, 1M-394
 mount request server — *mountd*, 1M-404
NFS security
 server for storing private encryption keys —
 keyserv, 1M-313
NFS, secure
 create a new key in publickey database —
 newkey, 1M-416
nfsstat — display NFS statistics, 1M-418
NIS
 binder process — *ypbind*, 1M-709
 make *ndbm* file — *makedbm*, 1M-359
 point binder process *ypbind* at a particular
 server — *ypset*, 1M-714
 return current version of a NIS map at a NIS

server

NIS, *continued*

- host — `yppoll`, 1M-713
- transfer NIS map from a NIS server to host — `ypxfr`, 1M-715

NIS database

- rebuild — `ypmake`, 1M-712

NIS+

- add /etc files and NIS maps into their corresponding NIS+ tables — `nisaddent`, 1M-426
- client and root server initialization utility — `nisinit`, 1M-434
- credential creation tool — `nisaddcred`, 1M-422
- display the contents of the NIS+ transaction log — `nislog`, 1M-437
- initialize a domain to store system administration information— `nissetup`, 1M-446
- `nisping` — ping NIS+ servers, 1M-438
- `nissetup` — initialize a NIS+ domain to serve clients, 1M-446
- `nisstat` — report NIS+ server statistics, 1M-448
- service daemon — `rpc.nisd`, 1M-526
- update public keys in NIS+ directory object — `nisupdkeys`, 1M-450
- utility to cache location information about NIS+ servers — `nis_cachemgr`, 1M-420
- utility to print out the contents of the shared cache file — `nisshowcache`, 1M-447

NIS+ credentials for NIS+ principals

- initialize — `nisclient`, 1M-430

NIS+ password update daemon

- `nisspasswd`, 1M-530
- `rpc.nisspasswd`, 1M-530

NIS+ servers

- set up — `nissserver`, 1M-444

- `nisaddcred` — create NIS+ credentials, 1M-422
- `nisaddent` — add /etc files and NIS maps into their corresponding NIS+ tables, 1M-426
- `nisclient` — initialize NIS+ credentials for NIS+ principals, 1M-430
- `nisspasswd` — NIS+ password update daemon,

- 1M-530
- `nisping` — ping NIS+ servers, 1M-438
- `nispopulate` — populate the NIS+ tables in a NIS+ domain., 1M-440
- `nissserver` — set up NIS+ servers, 1M-444
- `nissetup` — initialize a domain to serve clients, 1M-446
- `nisshowcache` — NIS+ utility to print out the contents of the shared cache file, 1M-447
- `nisstat` — report NIS+ server statistics, 1M-448
- `nscd` — name service cache daemon, 1M-457
- `nslookup` — query Internet domain name servers, 1M-459
- `nstest` — query Internet domain name servers, 1M-463
- `nulladm` — accounting shell procedure, 1M-31

O

- `obpsym` — Kernel Symbolic Debugging for Open-Boot Firmware, 1M-466
- Kernel Symbolic Name Syntax, 1M-466
- operating system
 - install SunOS/SVR4 — `suninstall`, 1M-636
 - remake the binary system and commands from source code — `mk`, 1M-360
 - save core dump — `savecore`, 1M-583
- operating system, bootable
 - load and execute — `boot`, 1M-87
- output system definition
 - display current — `sysdef`, 1M-653
- owner of file
 - change — `chown`, 1M-108

P

- packages
 - install, upgrade, and remove software — `swmtool`, 1M-640
- parse `clustertoc` file based on dynamic entries —
 - `parse_dynamic_clustertoc`, 1M-468
- `parse_dynamic_clustertoc` — parse `clustertoc` file based on dynamic entries, 1M-468
- `passmgmt` — password files management, 1M-469
- password file

password file, *continued*
 check for inconsistencies — pwck, 1M-506

passwords
 installs and updates /etc/shadow with information from /etc/passwd — pwconv, 1M-507
 management — passmgmt, 1M-469

pbind — control process bindings to processors, 1M-471

pcfs
 mount — mount_pcfs, 1M-398

PCMCIA user daemon — pcmciad, 1M-473

pcmciad — PCMCIA user daemon, 1M-473

per-processor statistics report — mpstat, 1M-405

perform surface analysis — diskscan, 1M-162

pfinstall — tests installation profiles, 1M-474

pkgadd — transfer software package to the system, 1M-478

pkgask — stores answers to an interactive software package, 1M-480

pkgchk — check accuracy of installation of a software package, 1M-481

pkgrm — removes package from the system, 1M-483

Point-to-Point Protocol
 — aspppd, 1M-51
 — aspppls, 1M-51

populate the NIS+ tables in a NIS+ domain — nispopulate, 1M-440

ports
 add or remove a service — pmadm, 1M-485
 create /dev entries and inittab entries for serial lines — ports, 1M-489
 enable or disable a service — pmadm, 1M-485
 install or replace a service configuration script — pmadm, 1M-485
 print requested service information — pmadm, 1M-485
 service administration — pmadm, 1M-485

poweroff — stop the processor, 1M-242

praudit — display audit trail, 1M-490

prctmp — accounting shell procedure, 1M-31

prdaily — accounting shell procedure, 1M-31

print queue
 accept or reject requests — accept, reject, 1M-20

print requests
 accept or reject — accept, reject, 1M-20

print service, LP, See LP print services

printer filters
 add and change — lpfilter, 1M-341
 list attributes — lpfilter, 1M-341
 remove — lpfilterer, 1M-341

printer forms
 add or change — lpforms, 1M-345
 delete — lpforms, 1M-348
 list attributes — lpforms, 1M-348
 listing the current alert — lpforms, 1M-350
 provide access — lpforms, 1M-348
 removing an alert definition — lpforms, 1M-351
 setting an alert to mount a form — lpforms, 1M-349
 terminating an active alert — lpforms, 1M-350

printers
 add and change printers — lpadmin, 1M-330
 define alerts for printer faults — lpadmin, 1M-330
 mount printer wheels — lpadmin, 1M-330
 remove printers — lpadmin, 1M-330
 set or change system default destination — lpadmin, 1M-330
 setting priorities — lpusers, 1M-356

process accounting
 — acctcms, 1M-24
 — accton, 1M-21
 — acctprc, 1M-29

process scheduler
 administration — dispadmin, 1M-163

processes
 initialization — init, 1M-289
 kill all active processes — killall, 1M-314
 using file or file structure — fuser, 1M-227

program, standalone
 load and execute — boot, 1M-87

programming tools

programming tools, *continued*

- install — install commands, 1M-120, 1M-159, 1M-293, 1M-295
- PROM monitor program — monitor, 1M-373
 - display and load program — eeprom, 1M-175
- prtacct — accounting shell procedure, 1M-31
- prtconf — print system configuration information, 1M-491
- prtdiag — print system diagnostic information, 1M-494
- prvtoc — print the VTOC of a block device, 1M-497
- psradm — set processors online or offline, 1M-499
- psrinfo — print processor information, 1M-500
- putdev — add, modify remove device entry from table, 1M-501
- putdgrp — add, remove and change device group table, 1M-504

Q

- quick halt
 - halt, 1M-242
- quota
 - remote quota daemon — rquotad, 1M-546
- quotas
 - check file system quota for inconsistencies — quotacheck, 1M-510
 - display a user's disk quota and usage — quota, 1M-509
 - displays number of blocks in filesystem currently owned by each user — quot, 1M-508
 - summarize quotas for a file system — repquota, 1M-518
 - turn off file system quotas — quotaoff, 1M-511
 - turn on file system quotas — quotaon, 1M-511
- quotas, user
 - edit — edquota, 1M-173

R

- RARP
 - server — in.rarpd, 1M-264
- re-preinstall — restore a system to its original factory condition, 1M-513
- real-time clock and GMT-lag management, provide
 - rtc, 1M-549
- reboot — restart the operating system, 1M-515
- reject — reject print requests, 1M-20
- remote execution server — in.rexecd, 1M-267
- remote login server — in.rlogind, 1M-269
 - rlogind, 1M-269
- remote magtape protocol server — rmt, 1M-521
- Remote Program Load Server, See rpld
- remote resources
 - mount NFS — mount_nfs, 1M-394
 - mount or unmount — mount, 1M-386
- remote system
 - execute remote command requests — uuxqt, 1M-698
 - make local resource unavailable for mounting
 - unshare, 1M-684
 - register with LP print service — lpsystem, 1M-353
 - set system date — rdate, 1M-512
 - shell server — in.rshd, 1M-274
 - user information server — in.fingerd, 1M-253
- remote system with debugging on, try to contact —
 - Uutry, 1M-19
 - uutry, 1M-19
- removable media mounter that automatically mounts a file system on a CD-ROM and floppy.
 - rmmount, 1M-519
- removef — remove a file from software database, 1M-517
- restore
 - incremental file system restore — ufsrestore, 1M-679
- return user identity — id, 1M-245
- Reverse Address Resolution Protocol, See RARP
- rlogind — remote login server, 1M-269
- rm_drv — remove a device driver from the system,

1M-516

`rm_install_client` — scripts used to install the Solaris software, 1M-296

`rmmount` — removable media mounter that automatically mounts a file system on a CD-ROM and floppy., 1M-519

`rmt` — remote magtape protocol server, 1M-521

root directory
 change for a command — `chroot`, 1M-109

`route` — manually manipulate routing tables, 1M-523

RPC
 kernel statistics server — `rpc.rstatd`, 1M-533
 network username server — `rpc.rusersd`, 1M-534
 network wall server — `rpc.rwall`, 1M-535
 NIS+ service daemon — `rpc.nisd`, 1M-526
 program number to universal addresses mapping — `rpcbind`, 1M-537
 remote execution server — `rpc.rexd`, 1M-532
 remote quota daemon — `rquotad`, 1M-546
 report information — `rpcinfo`, 1M-538
 sends one-way stream of packets to host — `spray`, 1M-618
 server which records packets sent by spray — `rpc.sprayd`, 1M-536
 server, Autofs mount/unmount requests — `automountd`, 1M-82
 server, NFS mount requests — `mountd`, 1M-404

RPC, secure
 create a new key in publickey database — `newkey`, 1M-416

`rpc.bootparamd` — boot parameter server, 1M-525

`rpc.nisd` — NIS+ service daemon, 1M-526
`rpc.nisd_resolv`, 1M-526, 1M-529

`rpc.nispasswd` — NIS+ password update daemon, 1M-530

`rpc.rexd` — remote execution daemon, 1M-532

`rpc.rstatd` — kernel statistics server, 1M-533

`rpc.rusersd` — network username server, 1M-534

`rpc.rwalld` — network wall server, 1M-535

`rpc.sprayd` — records packets sent by spray, 1M-536

`rpcbind` — converts RPC program numbers to universal addresses, 1M-537

`rpcinfo` — report RPC information, 1M-538

`rpld` — x86 Network Booting RPL (Remote Program Load) Server, 1M-541

`rquotad` — remote quota server, 1M-546

`rsh` — the restricted shell command interpreter

`rtc` — provide all real-time clock and GMT-lag management, 1M-549

`runacct` — accounting shell procedure, 1M-31, 1M-550

`rwall` — write to all users over a network, 1M-552

`rwall` daemon
 network wall server — `rpc.rwalld`, 1M-535

S

s5fs
 mount — `mount_s5fs`, 1M-399

`sac` — Service Access Controller (SAC), 1M-553

`sacadm` — Service Access Controller administration

`sadmind` — Distributed system administration daemon, 1M-559

`saf` — Service Access Facility (SAF), 1M-563

SAF
 format and output port monitor-specific information — `ttyadm`, 1M-664
 port monitor for terminal ports — `ttymon`, 1M-667

`savecore` — save OS core dump, 1M-583

scheduler, process
 administration — `dispadmin`, 1M-163

scripts used to install the Solaris software —
`install_scripts`, 1M-296
`add_install_client`, 1M-296
`check`, 1M-296
`rm_install_client`, 1M-296
`setup_install_server`, 1M-296

`sendmail` — mail delivery system, 1M-585

set serial line interface
 set operating parameters — `syncinit`, 1M-644

server_upgrade — upgrade clients of a heterogeneous OS server, 1M-594
Initial State of Solaris, 1M-595
State of sun4e clients, 1M-596
Upgrade the 2.4 OS server to 2.5, 1M-595
Upgrade the sun4L (Hardware Partner) client and services, 1M-596
Upgrade the x86 clients and services, 1M-595

servers

- automountd — moun/unmount request server, 1M-82
- biff server — in.comsat, 1M-252
- DARPA TELNET virtual terminal protocol server — in.telnetd, 1M-279
- DARPA trivial name server — in.tnamed, 1M-282
- in.rexecd — remote execution server, 1M-267
- in.uucpd — UUCP server, 1M-283
- inetd — Internet services daemon, 1M-284
- Internet domain name server — in.named, 1M-261
- mountd — mount request server, 1M-404
- RARP server — in.rarpd, 1M-264
- system status server — in.rwhod, 1M-276
- talk program server — in.talkd, 1M-278

servers, NIS+

- location information — nis_cachemgr, 1M-420

Service Access Controller, See SAC, See SAC
Service Access Facility, See SAF

setuname — changes machine information, 1M-598

setup_install_server — scripts used to install the Solaris software, 1M-296

share — make local resource available for mounting by remote systems, 1M-599

share_nfs — make local NFS filesystem available for mounting by remote systems, 1M-601

shareall — multiple resources, 1M-604

shell

- remote shell server — in.rshd, 1M-274

SHELL variable — rsh, 1M-547

shells

- the restricted shell command interpreter — rsh,

shells, *continued*
1M-547

showmount — display remote mounts, 1M-605

showrev — displays revision information for current hardware and software, 1M-606

shutacct — accounting shell procedure, 1M-31

shutdown — shutdown system, change system state, 1M-608

snoop — capture and inspect network packets, 1M-610

software package

- add a file to the software installation database — installf, 1M-300
- check accuracy of installation — pkgchk, 1M-481
- remove a file from software database — removef, 1M-517
- removes package from the system — pkgrm, 1M-483
- stores answers to an interactive package — pkgask, 1M-480

SPARCstorage Array

- administration program — ssaadm, 1M-619
- command line interface — ssacli, 1M-622

spray — sends one-way stream of packets to host, 1M-618

ssaadm — Administration program for the SPARCstorage Array, 1M-619

ssacli — SPARCstorage Array Command Line Interface, 1M-622

startup — accounting shell procedure, 1M-31

startup procedures — boot, 1M-299

statd — network status monitor, 1M-625

statistics

- audit — auditstat, 1M-74
- for file system — ff, 1M-186
- for ufs file system — ff, 1M-188
- I/O — iostat, 1M-303
- NFS, display — nfsstat, 1M-418
- per-processor — mpstat, 1M-405

stop the processor — halt, 1M-242
poweroff, 1M-242

store mail in a mailbox — mail.local, 1M-358

STREAMS

automatically pushed modules — `autopush`, 1M-83

detach a name from file descriptor — `fdetach`, 1M-180

error logger cleanup program — `strclean`, 1M-628

error logger daemon — `strerr`, 1M-629

port monitor for terminal ports — `ttymon`, 1M-667

print trace messages — `strace`, 1M-626

`su` — become super-user or another user, 1M-633

`sulogin` — single-user mode automatically invoked when system is first started, 1M-635

`suninstall` — install SunOS/SVR4, 1M-636

super-user command — `su`, 1M-633

`sulogin` — single-user mode automatically invoked when system is first started, 1M-635

`swap` — administer the system swap areas, 1M-637

`swmtool` — install, upgrade and remove software packages, 1M-640

`sxconfig` — configure contiguous memory for `sx` video, 1M-641

`sync` — update super block, 1M-643

synchronous serial link

- report driver statistics — `syncstat`, 1M-649

synchronous serial loopback test program — `syncloop`, 1M-646

`syncinit` — set serial line interface operating parameters., 1M-644

`syncloop` — synchronous serial loopback test program, 1M-646

`syncstat` — report driver statistics from a synchronous serial link., 1M-649

`sys-unconfig` — undo system configuration, 1M-651

`sysdef` — displays current system definition, 1M-653

`sysidconfig` — execute or define configuration applications, 1M-655

`sysidnet` — system configuration, 1M-657

`sysidnis` — system configuration, 1M-657

`sysidroot` — system configuration, 1M-657

`sysidsys` — system configuration, 1M-657

`sysidtool` — system configuration, 1M-657

`syslogd` — system log message daemon, 1M-659

system activity reports — `sar`, 1M-581

system administration

- access control lists — `admintool`, 1M-39
- add a file to the software installation database — `installf`, 1M-300
- administer hosts on a network — `tyhstmgr`, 1M-666
- analyze the SunOS 4.x client support after converting to Solaris 2.x — `discover4x`, 1M-159
- control for basic administrative functions — `uadmin`, 1M-672
- converts SunOS 4.x clients to work from Solaris 2.x servers — `convert4x`, 1M-120
- create cat files for manual pages — `catman`, 1M-101
- get entries from administrative database — `getent`, 1M-234
- install commands — `install`, 1M-293
- install commands — `install4x`, 1M-295
- NIS+ aliases — `aliasadm`, 1M-41
- system databases — `admintool`, 1M-39

system configuration — `sysidtool`, 1M-657

- print information — `prtconf`, 1M-491

`sysidnet`, 1M-657

`sysidnis`, 1M-657

`sysidroot`, 1M-657

`sysidsys`, 1M-657

- undo — `sys_unconfig`, 1M-651

system definition

- display current — `sysdef`, 1M-653

system diagnostic

- print information — `prtdiag`, 1M-494

system diagnostic messages

- error log — `dmesg`, 1M-165

system information

- displays revision information for current hardware and software — `showrev`, 1M-606

system log daemon — `syslog`, 1M-659

system memory

system memory, *continued*
 examine images — `crash`, 1M-121

system parameters
 change value — `setuname`, 1M-598

System Performance Analysis Utilities Tools
 system activity report package — `sar`, 1M-581

system PROM monitor program — `monitor`,
 1M-373

system shutdown
 — `halt`, 1M-242

system status server — `in.rwhod`, 1M-276

T

tapes — adds `/dev` entries for tape drives attached
 to the system, 1M-660

TCP/IP
 File Transfer Protocol daemon on remote host
 — `in.ftpd`, 1M-254

`tcxconfig` — configure the default linearity of the
 24-bit TrueColor Visual for OpenWindows with
 an S24 frame buffer system (TCX), 1M-662

`telinit` — process control initialization, 1M-289

terminal
 compare or print terminfo descriptions —
 `infocmp`
 set type, modes, speed, and line discipline —
 `getty`, 1M-236

terminfo
 compiler — `tic`, 1M-663

test installation profiles — `pinstall`, 1M-474

time zone
 compiler — `zic`, 1M-718
 print current time in each named on command
 line — `zdump`, 1M-717

timed event services
 daemon for cron — `cron`, 1M-131

`tmpfs`
 mount — `mount_tmpfs`, 1M-400

trace messages
 STREAMS — `strace`, 1M-626

transfer software package to the system — `pkgadd`,
 1M-478

TTY ports

TTY ports, *continued*
 maintain line settings and hunt sequences —
 `sttydefs`, 1M-631

`ttyadm` — format and output port monitor-specific
 information, 1M-664

`ttyhstmgr` — administer hosts on a network,
 1M-666

`ttymon` — port monitor for terminal ports, 1M-667

`tunefs` — tuneup an existing file system, 1M-670

`turnacct` — accounting shell procedure, 1M-31

U

`ufs`
 check and repair — `fsck_ufs`, 1M-213
 construct file system — `mkfs_ufs`, 1M-366
 display amount of disk space — `df_ufs`,
 1M-153
 generate a list of pathnames vs i-numbers —
 `ncheck_ufs`, 1M-408
 incremental file system restore — `ufsre-`
 `store`, 1M-679
 make image copy of file system—
 `volcopy_ufs`, 1M-703
 mount — `mount_ufs`, 1M-401
 provide labels for ufs — `labelit_ufs`,
 1M-318

`ufs` file system debugger — `fsdb_ufs`, 1M-217

`ufsdump` — incremental file system dump, 1M-673

`umount` — unmount file systems and remote
 resources, 1M-386

`umountall` — unmount multiple file systems,
 1M-403

unconfigure
 undo system configuration — `sys_unconfig`,
 1M-651

UNIX system executable file containing basic
 operating system services — `kernel`, 1M-311

unmount
 establish table — `setmnt`, 1M-597

`unshare` — make local resource unavailable for
 mounting by remote systems, 1M-684

`unshare_nfs` — make local NFS filesystem una-
 vailable for mounting by remote systems,

1M-685
 unshareall — multiple resources, 1M-604
 update super block — sync, 1M-643
 upgrade clients of a heterogeneous OS server —
 server_upgrade, 1M-594
 user IDs
 administer a new user login on the system —
 useradd, 1M-686
 become super-user or another user — su,
 1M-633
 user information
 server, remote — in.fingerd, 1M-253
 user quotas
 remote quota daemon — rquotad, 1M-546
 useradd — administer a new user login on the sys-
 tem, 1M-686
 userdel — delete a user's login from the system,
 1M-689
 usermod — modify a user's login information on
 the system, 1M-690
 users
 list login information — logins, 1M-329
 who is doing what — whodo, 1M-707
 utmp2wtmp — create an entry for each user logged
 on in file /var/adm/wtmp, 1M-21
 uuccheck — check the uucp directories and permis-
 sions file, 1M-692
 uucico — file transport program for uucp system,
 1M-693
 uucleanup — uucp spool directory clean-up,
 1M-695
 uucp
 check directories and permissions file —
 uuccheck, 1M-692
 clean-up of spool directory — uucleanup,
 1M-695
 file transport program — uucico, 1M-693
 scheduler for file transport program —
 uusched, 1M-697
 server — in.uucpd, 1M-283
 uusched — scheduler for uucp transport program,
 1M-697
 Uutry — try to contact remote system with debug-

ging on, 1M-19
 uuxqt — execute remote command requests,
 1M-698

V

virtual memory
 display statistics — vmstat, 1M-699
 vmstat — display virtual memory statistics,
 1M-699
 vold — Volume Management daemon to manage
 CD-ROM and floppy devices, 1M-704
 Volume Management
 floppy and CD-ROM management daemon —
 vold, 1M-704
 removable media mounter — rmmount,
 1M-519
 vtoc
 print VTOC of a block device — prtvtoc,
 1M-497

W

wall — write to all users, 1M-706
 wchrtbl — generate wide character classification
 tables, 1M-110
 whodo — who is doing what, 1M-707
 wtmpfix — correct connect accounting records
 date/time stamp, 1M-228

X

x86 Network Booting RPL (Remote Program Load)
 Server — rpld, 1M-541

Y

YP
 make ndbm file — makedbm, 1M-359
 ypbind — NIS binder process, 1M-709
 ypinit — set up NIS client, 1M-711
 ypmake — rebuild NIS database, 1M-712
 yppoll — return current version of a NIS map at a
 NIS server host, 1M-713
 ypset — point NIS binder process ypbind at a
 particular server, 1M-714

`ypxfr` — transfer NIS map from a NIS server to
host, 1M-715

Z

`zdump` — prints the current time in each zonename
named on command line, 1M-717

`zic` — time zone compiler, 1M-718