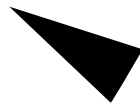


# SunOS Reference Manual

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



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

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

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

Sun, Sun Microsystems, the Sun logo, SunSoft, Solaris, SunOS, OpenWindows, DeskSet, ONC, ONC+, and NFS are trademarks, or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

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

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

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

---

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

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

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

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

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

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



# *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.

## *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 other 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.

## *ATTRIBUTES*

This section lists characteristics of commands, utilities, and device drivers by defining the attribute type and its corresponding value. (See **attributes(5)** for more information.)

## *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.



<b>NAME</b>	Intro, intro – introduction to file formats
<b>DESCRIPTION</b>	<p>This section outlines the formats of various files. The C structure declarations for the file formats are given where applicable. Usually, the headers containing these structure declarations can be found in the directories <b>/usr/include</b> or <b>/usr/include/sys</b>. For inclusion in C language programs, however, the syntax <b>#include &lt;filename.h&gt;</b> or <b>#include &lt;sys/filename.h&gt;</b> should be used.</p> <p>Because the operating system now allows the existence of multiple file system types, there are several instances of multiple manual pages with the same name. These pages all display the name of the FSType to which they pertain, in the form <code>name_ <i>fstype</i></code> at the top of the page. For example, <b>fs_ufs(4)</b>.</p>
<b>INTERFACES</b>	<p>Descriptions of shared objects may include a definition of the global symbols that define the shared objects' public interface, for example <b>SUNW_1.1</b>. Other interfaces may exist within the shared object, for example <b>SUNW_private.1.1</b>. The public interface provides a stable, committed set of symbols for application development. The private interfaces are for internal use only, and may change at any time.</p> <p>For many shared objects, an archive library is provided for backward compatibility. Use of these libraries may restrict an applications ability to migrate between different Solaris releases. As dynamic linking is the preferred compilation method on Solaris, the use of these libraries is discouraged.</p>

<b>Name</b>	<b>Description</b>
<b>acct(4)</b>	per-process accounting file format
<b>addresses(4)</b>	See <b>aliases(4)</b>
<b>admin(4)</b>	installation defaults file
<b>aliases(4)</b>	addresses and aliases for sendmail
<b>a.out(4)</b>	Executable and Linking Format (ELF) files
<b>ar(4)</b>	archive file format
<b>archives(4)</b>	device header
<b>asetenv(4)</b>	ASET environment file
<b>asetmasters(4)</b>	ASET master files
<b>audit_class(4)</b>	audit class definitions
<b>audit_control(4)</b>	control information for system audit daemon
<b>audit_data(4)</b>	current information on audit daemon
<b>audit_event(4)</b>	audit event definition and class mapping
<b>audit.log(4)</b>	audit trail file
<b>audit_user(4)</b>	per-user auditing data file

<b>bootparams(4)</b>	boot parameter data base
<b>cdtoc(4)</b>	CD-ROM table of contents file
<b>cklist.high(4)</b>	See <b>asetmasters(4)</b>
<b>cklist.low(4)</b>	See <b>asetmasters(4)</b>
<b>cklist.med(4)</b>	See <b>asetmasters(4)</b>
<b>clustertoc(4)</b>	cluster table of contents description file
<b>compver(4)</b>	compatible versions file
<b>copyright(4)</b>	copyright information file
<b>core(4)</b>	core image file
<b>default_fs(4)</b>	specify the default file system type for local or remote file systems
<b>defaultrouter(4)</b>	configuration file for default router(s)
<b>depend(4)</b>	software dependencies file
<b>device_allocate(4)</b>	device_allocate file
<b>device.cinfo(4)</b>	devconfig configuration files
<b>device_maps(4)</b>	device_maps file
<b>dfstab(4)</b>	file containing commands for sharing resources across a network
<b>dhcp(4)</b>	file containing default parameter values for the location and type of the databases used by the DHCP service
<b>dhcp_network(4)</b>	dhcp network DHCP database
<b>dhcptab(4)</b>	DHCP configuration parameter table
<b>dhcptags(4)</b>	DHCP option mnemonic mapping table
<b>dialups(4)</b>	list of terminal devices requiring a dial-up password
<b>dir(4)</b>	See <b>dir_ufs(4)</b>
<b>dirent(4)</b>	file system independent directory entry
<b>dir_ufs(4)</b>	format of ufs directories
<b>d_passwd(4)</b>	dial-up password file
<b>driver.conf(4)</b>	driver configuration files
<b>dumpdates(4)</b>	See <b>ufsdump(4)</b>
<b>eisa(4)</b>	See <b>sysbus(4)</b>
<b>environ(4)</b>	user-preference variables files for AT&T FACE
<b>ethers(4)</b>	Ethernet address to hostname database or domain
<b>fbtab(4)</b>	See <b>logindevperm(4)</b>
<b>fd(4)</b>	file descriptor files
<b>filehdr(4)</b>	file header for common object files

<b>format.dat(4)</b>	disk drive configuration for the format command
<b>forward(4)</b>	See <b>aliases(4)</b>
<b>fs(4)</b>	See <b>default_fs(4)</b>
<b>fspec(4)</b>	format specification in text files
<b>fstypes(4)</b>	file that registers distributed file system packages
<b>fs_ufs(4)</b>	format of a ufs file system volume
<b>group(4)</b>	group file
<b>holidays(4)</b>	prime/nonprime table for the accounting system
<b>hosts(4)</b>	host name database
<b>hosts.equiv(4)</b>	trusted remote hosts and users
<b>inetd.conf(4)</b>	Internet servers database
<b>init.d(4)</b>	initialization and termination scripts for changing init states
<b>inittab(4)</b>	script for init
<b>inode(4)</b>	See <b>fs_ufs(4)</b>
<b>inode_ufs(4)</b>	See <b>fs_ufs(4)</b>
<b>isa(4)</b>	See <b>sysbus(4)</b>
<b>issue(4)</b>	issue identification file
<b>keytables(4)</b>	keyboard table descriptions for loadkeys and dumpkeys
<b>krb.conf(4)</b>	Kerberos configuration file
<b>krb.realms(4)</b>	host to Kerberos realm translation file
<b>libadm(4)</b>	general administrative library
<b>libaio(4)</b>	the asynchronous I/O library
<b>libbsm(4)</b>	basic security library
<b>libc(4)</b>	the C library
<b>libci(4)</b>	Sun Solstice Enterprise Agent Component Interface Library
<b>libdevicid(4)</b>	device id library
<b>libdl(4)</b>	the dynamic linking interface library
<b>libdmi(4)</b>	Sun Solstice Enterprise Agent DMI Library
<b>libdmimi(4)</b>	Sun Solstice Enterprise Agent Management Interface Library
<b>libelf(4)</b>	ELF access library
<b>libintl(4)</b>	internationalization library
<b>libkrb(4)</b>	Kerberos library

<b>libkstat(4)</b>	kernel statistics library
<b>libkvm(4)</b>	Kernel Virtual Memory access library
<b>libmapmalloc(4)</b>	an alternative memory allocator library
<b>libmp(4)</b>	multiple precision library
<b>libnisdb(4)</b>	NIS+ Database access library
<b>libnsl(4)</b>	the network services library
<b>libpam(4)</b>	interface library for PAM (Pluggable Authentication Module)
<b>libposix4(4)</b>	POSIX.1b Realtime Extensions library
<b>libpthread(4)</b>	POSIX threads library
<b>librac(4)</b>	remote asynchronous calls library
<b>libresolv(4)</b>	resolver library
<b>librpcsoc(4)</b>	obsolete RPC library
<b>librpcsvc(4)</b>	miscellaneous RPC services library
<b>libsec(4)</b>	File Access Control List library
<b>libsocket(4)</b>	the sockets library
<b>libssagent(4)</b>	Sun Solstice Enterprise Agent Library
<b>libssasnmplib(4)</b>	Sun Solstice Enterprise SNMP Library
<b>libsys(4)</b>	the system library
<b>libthread(4)</b>	the threads library
<b>libthread_db(4)</b>	threads debugging library
<b>libtnfctl(4)</b>	library of TNF probe control routines for use by processes and the kernel
<b>libucb(4)</b>	the UCB compatibility library
<b>libvolmgt(4)</b>	volume management library
<b>libw(4)</b>	the wide character library
<b>libxfn(4)</b>	the XFN interface library
<b>libxnet(4)</b>	X/Open Networking Interfaces library
<b>limits(4)</b>	header for implementation-specific constants
<b>loadfont(4)</b>	format of a font file used as input to the loadfont utility
<b>logindevperm(4)</b>	login-based device permissions
<b>loginlog(4)</b>	log of failed login attempts
<b>magic(4)</b>	file command's magic number file
<b>mca(4)</b>	See <b>sysbus(4)</b>
<b>mnttab(4)</b>	mounted file system table
<b>netconfig(4)</b>	network configuration database

<b>netgroup(4)</b>	list of network groups
<b>netid(4)</b>	netname database
<b>netmasks(4)</b>	network mask database
<b>netrc(4)</b>	file for ftp remote login data
<b>networks(4)</b>	network name database
<b>nisfiles(4)</b>	NIS+ database files and directory structure
<b>nologin(4)</b>	message displayed to users attempting to log on in the process of a system shutdown
<b>note(4)</b>	specify legal annotations
<b>nscd.conf(4)</b>	name service cache daemon configuration
<b>nsswitch.conf(4)</b>	configuration file for the name service switch
<b>order(4)</b>	package installation order description file
<b>ott(4)</b>	FACE object architecture information
<b>packagetoc(4)</b>	package table of contents description file
<b>packingrules(4)</b>	packing rules file for cachefs and filesync
<b>pam.conf(4)</b>	configuration file for pluggable authentication modules
<b>passwd(4)</b>	password file
<b>pathalias(4)</b>	alias file for FACE
<b>path_to_inst(4)</b>	device instance number file
<b>pci(4)</b>	configuration files for PCI device drivers
<b>pcmcia(4)</b>	PCMCIA nexus driver
<b>phones(4)</b>	remote host phone number database
<b>pkginfo(4)</b>	package characteristics file
<b>pkgmap(4)</b>	package contents description file
<b>platform(4)</b>	directory of files specifying supported platforms
<b>plot(4B)</b>	graphics interface
<b>power.conf(4)</b>	power management configuration information file
<b>pref(4)</b>	See <b>environ(4)</b>
<b>printers(4)</b>	user-configurable printer alias database
<b>printers.conf(4)</b>	system printing configuration database
<b>proc(4)</b>	/proc, the process file system
<b>profile(4)</b>	setting up an environment for user at login time
<b>protocols(4)</b>	protocol name database
<b>prototype(4)</b>	package information file
<b>pseudo(4)</b>	configuration files for pseudo device drivers

<b>publickey(4)</b>	public key database
<b>queuedefs(4)</b>	queue description file for at, batch, and cron
<b>remote(4)</b>	remote host description file
<b>resolv.conf(4)</b>	configuration file for name server routines
<b>rhosts(4)</b>	See <b>hosts.equiv(4)</b>
<b>rmmount.conf(4)</b>	removable media mounter configuration file
<b>rmtab(4)</b>	remote mounted file system table
<b>route(4)</b>	kernel packet forwarding database
<b>routing(4)</b>	system support for packet network routing
<b>rpc(4)</b>	rpc program number data base
<b>rpld.conf(4)</b>	Remote Program Load (RPL) server configuration file
<b>rt_dptbl(4)</b>	real-time dispatcher parameter table
<b>sbus(4)</b>	configuration files for SBus device drivers
<b>sccsfile(4)</b>	format of an SCCS history file
<b>scsi(4)</b>	configuration files for SCSI target drivers
<b>securenets(4)</b>	configuration file for NIS security
<b>services(4)</b>	Internet services and aliases
<b>shadow(4)</b>	shadow password file
<b>sharetab(4)</b>	shared file system table
<b>shells(4)</b>	shell database
<b>sock2path(4)</b>	file that maps sockets to transport providers
<b>space(4)</b>	disk space requirement file
<b>sulog(4)</b>	su command log file
<b>sysbus(4)</b>	device tree properties for ISA, EISA, and MCA bus device drivers
<b>sysidcfg(4)</b>	system identification configuration file
<b>syslog.conf(4)</b>	configuration file for syslogd system log daemon
<b>system(4)</b>	system configuration information file
<b>telnetrc(4)</b>	file for telnet default options
<b>term(4)</b>	format of compiled term file
<b>terminfo(4)</b>	terminal and printer capability database
<b>timezone(4)</b>	default timezone data base
<b>TIMEZONE(4)</b>	set default system time zone and locale
<b>tnf_kernel_probes(4)</b>	TNF kernel probes
<b>ts_dptbl(4)</b>	time-sharing dispatcher parameter table

<b>ttydefs(4)</b>	file contains terminal line settings information for ttymon
<b>ttysrch(4)</b>	directory search list for ttyname
<b>tune.high(4)</b>	See <b>asetmasters(4)</b>
<b>tune.low(4)</b>	See <b>asetmasters(4)</b>
<b>tune.med(4)</b>	See <b>asetmasters(4)</b>
<b>ufsdump(4)</b>	incremental dump format
<b>uid_aliases(4)</b>	See <b>asetmasters(4)</b>
<b>unistd(4)</b>	header for symbolic constants
<b>updaters(4)</b>	configuration file for NIS updating
<b>utmp(4)</b>	utmp and wtmp entry formats
<b>utmpx(4)</b>	utmpx and wtmpx entry formats
<b>variables(4)</b>	See <b>environ(4)</b>
<b>vfstab(4)</b>	table of file system defaults
<b>vme(4)</b>	configuration files for VMEbus device drivers
<b>vold.conf(4)</b>	Volume Management configuration file
<b>wtmp(4)</b>	See <b>utmp(4)</b>
<b>wtmpx(4)</b>	See <b>utmpx(4)</b>
<b>ypfiles(4)</b>	Network Information Service Version 2, formerly known as YP

<b>NAME</b>	acct – per-process accounting file format
<b>SYNOPSIS</b>	<b>#include</b> <sys/types.h> <b>#include</b> <sys/acct.h>
<b>DESCRIPTION</b>	<p>Files produced as a result of calling <b>acct(2)</b> have records in the form defined by &lt;sys/acct.h&gt;, whose contents are:</p> <pre> <b>typedef</b> <b>ushort</b> <b>comp_t</b>;          /* pseudo "floating point" representation */                                    /* 3 bit base-8 exponent in the high */                                    /* order bits, and a 13-bit fraction */                                    /* in the low order bits. */  <b>struct</b> <b>acct</b> {     <b>char</b>    <b>ac_flag</b>;          /* Accounting flag */     <b>char</b>    <b>ac_stat</b>;          /* Exit status */     <b>uid_t</b>   <b>ac_uid</b>;           /* Accounting user ID */     <b>gid_t</b>   <b>ac_gid</b>;           /* Accounting group ID */     <b>dev_t</b>   <b>ac_tty</b>;           /* control tty */     <b>time_t</b>  <b>ac_btime</b>;         /* Beginning time */     <b>comp_t</b>  <b>ac_utime</b>;         /* accounting user time in clock */                                    /* ticks */     <b>comp_t</b>  <b>ac_stime</b>;         /* accounting system time in clock */                                    /* ticks */     <b>comp_t</b>  <b>ac_etime</b>;         /* accounting total elapsed time in clock */                                    /* ticks */     <b>comp_t</b>  <b>ac_mem</b>;           /* memory usage in clicks (pages) */     <b>comp_t</b>  <b>ac_io</b>;            /* chars transferred by read/write */     <b>comp_t</b>  <b>ac_rw</b>;            /* number of block reads/writes */     <b>char</b>    <b>ac_comm[8]</b>;       /* command name */ };  /*  * Accounting Flags  */ <b>#define</b> <b>AFORK</b>  <b>01</b>           /* has executed fork, but no exec */ <b>#define</b> <b>ASU</b>    <b>02</b>           /* used super-user privileges */ <b>#define</b> <b>ACCTF</b>  <b>0300</b>         /* record type */ <b>#define</b> <b>AEXPND</b> <b>040</b>          /* Expanded Record Type – default */ </pre> <p>In <b>ac_flag</b>, the <b>AFORK</b> flag is turned on by each <b>fork</b> and turned off by an <b>exec</b>. The <b>ac_comm</b> field is inherited from the parent process and is reset by any <b>exec</b>. Each time the system charges the process with a clock tick, it also adds to <b>ac_mem</b> the current process size, computed as follows:</p> $(data\ size) + (text\ size) / (number\ of\ in-core\ processes\ using\ text)$



The value of `ac_mem / (ac_stime + ac_untime)` can be viewed as an approximation to the mean process size, as modified by text sharing.

The structure `tacct`, (which resides with the source files of the accounting commands), represents a summary of accounting statistics for the user id `ta_uid`. This structure is used by the accounting commands to report statistics based on user id.

```

/*
 * total accounting (for acct period), also for day
 */
struct  tacct {
    uid_t      ta_uid;          /* user id */
    char       ta_name[8];     /* login name */
    float      ta_cpu[2];      /* cum. cpu time in minutes, */
                                /* p/np (prime/non-prime time) */
    float      ta_kcore[2];    /* cum. kcore-minutes, p/np */
    float      ta_con[2];      /* cum. connect time in minutes, */
                                /* p/np */
    float      ta_du;          /* cum. disk usage (blocks)*/
    long       ta_pc;          /* count of processes */
    unsigned short ta_sc;      /* count of login sessions */
    unsigned short ta_dc;      /* count of disk samples */
    unsigned short ta_fee;     /* fee for special services */
};

```

`ta_cpu`, `ta_kcore`, and `ta_con` contain usage information pertaining to prime time and non-prime time hours. The first element in each array represents the time the resource was used during prime time hours. The second element in each array represents the time the resource was used during non-prime time hours. Prime time and non-prime time hours may be set in the `holidays` file (see `holidays(4)`).

`ta_kcore` is a cumulative measure of the amount of memory used over the accounting period by processes owned by the user with uid `ta_uid`. The amount shown represents kilobyte segments of memory used, per minute.

`ta_con` represents the amount of time the user was logged in to the system.

**FILES** /etc/acct/holidays prime/non-prime time table

**SEE ALSO** acctcom(1), acct(1M), acctcon(1M), acctmerg(1M), acctprc(1M), acctsh(1M), prtacct(1M), runacct(1M), shutacct(1M), acct(2), exec(2), fork(2)

**NOTES** The `ac_mem` value for a short-lived command gives little information about the actual size of the command, because `ac_mem` may be incremented while a different command (for example, the shell) is being executed by the process.

<b>NAME</b>	admin – installation defaults file
<b>DESCRIPTION</b>	<p><i>admin</i> is a generic name for an ASCII file that defines default installation actions by assigning values to installation parameters. For example, it allows administrators to define how to proceed when the package being installed already exists on the system.</p> <p><i>/var/sadm/install/admin/default</i> is the default <i>admin</i> file delivered with this release. The default file is not writable, so to assign values different from this file, create a new <i>admin</i> file. There are no naming restrictions for <i>admin</i> files. Name the file when installing a package with the <b>-a</b> option of <b>pkgadd(1M)</b>. If the <b>-a</b> option is not used, the default <i>admin</i> file is used.</p> <p>Each entry in the <i>admin</i> file is a line that establishes the value of a parameter in the following form:</p> <p style="text-align: center;"><i>param=value</i></p> <p>Eleven parameters can be defined in an <i>admin</i> file, but it is not required to assign values to all eleven parameters. If a value is not assigned, <b>pkgadd(1M)</b> asks the installer how to proceed.</p> <p>The eleven parameters and their possible values are shown below except as noted. They may be specified in any order. Any of these parameters (except the <b>mail</b> parameter) can be assigned the value <b>ask</b>, which means that if the situation occurs the installer is notified and asked to supply instructions at that time (see <b>NOTES</b>).</p> <p><b>basedir</b>            Indicates the base directory where relocatable packages are to be installed. If there is no <b>basedir</b> entry in the file, the installer will be prompted for a path name, as if the file contained the entry <b>basedir=ask</b>. This parameter can also be set to <b>default</b> (entry is <b>basedir=default</b>). In this instance, the package is installed into the base directory specified by the <b>BASEDIR</b> parameter in the <b>pkginfo(4)</b> file.</p> <p><b>mail</b>                Defines a list of users to whom mail should be sent following installation of a package. If the list is empty, no mail is sent. If the parameter is not present in the <i>admin</i> file, the default value of <b>root</b> is used. The <b>ask</b> value cannot be used with this parameter.</p> <p><b>runlevel</b>           Indicates resolution if the run level is not correct for the installation or removal of a package. Options are:</p> <p style="padding-left: 40px;"><b>nocheck</b>        Do not check for run level.</p> <p style="padding-left: 40px;"><b>quit</b>            Abort installation if run level is not met.</p> <p><b>conflict</b>           Specifies what to do if an installation expects to overwrite a previously installed file, thus creating a conflict between packages. Options are:</p> <p style="padding-left: 40px;"><b>nocheck</b>        Do not check for conflict; files in conflict will be overwritten.</p> <p style="padding-left: 40px;"><b>quit</b>            Abort installation if conflict is detected.</p> <p style="padding-left: 40px;"><b>nochange</b>        Override installation of conflicting files; they will not</p>

	be installed.
<b>setuid</b>	Checks for executables which will have setuid or setgid bits enabled after installation. Options are: <ul style="list-style-type: none"> <li><b>nocheck</b> Do not check for setuid executables.</li> <li><b>quit</b> Abort installation if setuid processes are detected.</li> <li><b>nochange</b> Override installation of setuid processes; processes will be installed without setuid bits enabled.</li> </ul>
<b>action</b>	Determines if action scripts provided by package developers contain possible security impact. Options are: <ul style="list-style-type: none"> <li><b>nocheck</b> Ignore security impact of action scripts.</li> <li><b>quit</b> Abort installation if action scripts may have a negative security impact.</li> </ul>
<b>partial</b>	Checks to see if a version of the package is already partially installed on the system. Options are: <ul style="list-style-type: none"> <li><b>nocheck</b> Do not check for a partially installed package.</li> <li><b>quit</b> Abort installation if a partially installed package exists.</li> </ul>
<b>instance</b>	Determines how to handle installation if a previous version of the package (including a partially installed instance) already exists. Options are: <ul style="list-style-type: none"> <li><b>quit</b> Exit without installing if an instance of the package already exists (does not overwrite existing packages).</li> <li><b>overwrite</b> Overwrite an existing package if only one instance exists. If there is more than one instance, but only one has the same architecture, it overwrites that instance. Otherwise, the installer is prompted with existing instances and asked which to overwrite.</li> <li><b>unique</b> Do not overwrite an existing instance of a package. Instead, a new instance of the package is created. The new instance will be assigned the next available instance identifier.</li> </ul>
<b>idepend</b>	Controls resolution if other packages depend on the one to be installed. Options are: <ul style="list-style-type: none"> <li><b>nocheck</b> Do not check package dependencies.</li> <li><b>quit</b> Abort installation if package dependencies are not met.</li> </ul>

**rdepend** Controls resolution if other packages depend on the one to be removed.  
Options are:

- nocheck** Do not check package dependencies.
- quit** Abort removal if package dependencies are not met.

**space** Controls resolution if disk space requirements for package are not met.  
Options are:

- nocheck** Do not check space requirements (installation fails if it runs out of space).
- quit** Abort installation if space requirements are not met.

**EXAMPLES**

Below is a sample *admin* file.

```
basedir=default
runlevel=quit
conflict=quit
setuid=quit
action=quit
partial=quit
instance=unique
idepend=quit
rdepend=quit
space=quit
```

**SEE ALSO**

**pkgadd(1M)**, **pkginfo(4)**

**NOTES**

The value **ask** should not be defined in an *admin* file that will be used for non-interactive installation (since by definition, there is no installer interaction). Doing so causes installation to fail when input is needed.

<b>NAME</b>	aliases, addresses, forward – addresses and aliases for sendmail
<b>SYNOPSIS</b>	<pre> /etc/mail/aliases /etc/mail/aliases.dir /etc/mail/aliases.pag ~/forward </pre>
<b>DESCRIPTION</b>	<p>These files contain mail addresses or aliases, recognized by <b>sendmail(1M)</b> for the local host:</p> <p><b>/etc/passwd</b> Mail addresses (usernames) of local users.</p> <p><b>/etc/mail/aliases</b> Aliases for the local host, in ASCII format. This file can be edited to add, update, or delete local mail aliases.</p> <p><b>/etc/mail/aliases. {dir , pag}</b> The aliasing information from <b>/etc/mail/aliases</b>, in binary, <b>dbm</b> format for use by <b>sendmail(1M)</b>. The program <b>newaliases(1)</b>, which is invoked automatically by <b>sendmail(1M)</b>, maintains these files.</p> <p><b>~/forward</b> Addresses to which a user's mail is forwarded (see <b>Automatic Forwarding</b>, below).</p> <p>In addition, the NIS name services aliases map <i>mail.aliases</i>, and the NIS+ <i>mail_aliases</i> table, both contain addresses and aliases available for use across the network.</p> <p><b>Addresses</b> As distributed, <b>sendmail(1M)</b> supports the following types of addresses:</p> <p><b>Local Usernames</b> <i>username</i> Each local <i>username</i> is listed in the local host's <b>/etc/passwd</b> file.</p> <p><b>Local Filenames</b> <i>pathname</i> Messages addressed to the absolute <i>pathname</i> of a file are appended to that file.</p> <p><b>Commands</b>   <i>command</i> If the first character of the address is a vertical bar (   ), <b>sendmail(1M)</b> pipes the message to the standard input of the <i>command</i> the bar precedes.</p> <p><b>DARPA-standard Addresses</b> <i>username@domain</i> If <i>domain</i> does not contain any '.' (dots), then it is interpreted as the name of a host in the current domain. Otherwise, the message is passed to a <i>mailhost</i> that determines how to get to the specified domain. Domains are divided into subdomains separated by dots, with the top-level domain on the right. Top-level domains include:</p> <ul style="list-style-type: none"> <li>.COM Commercial organizations.</li> <li>.EDU Educational organizations.</li> <li>.GOV Government organizations.</li> </ul>

.MIL Military organizations.

For example, the full address of John Smith could be:

**js@jsmachine.Podunk-U.EDU**

if he uses the machine named **jsmachine** at Podunk University.

#### uucp Addresses

... [host!]host!username

These are sometimes mistakenly referred to as “Usenet” addresses. **uucp**(1C) provides links to numerous sites throughout the world for the remote copying of files.

Other site-specific forms of addressing can be added by customizing the **sendmail.cf** configuration file. See **sendmail**(1M) for details. Standard addresses are recommended.

#### Aliases Local Aliases

**/etc/mail/aliases** is formatted as a series of lines of the form

*aliasname:address[, address]*

*aliasname* is the name of the alias or alias group, and *address* is the address of a recipient in the group. Aliases can be nested. That is, an *address* can be the name of another alias group. Because of the way **sendmail**(1M) performs mapping from upper-case to lower-case, an *address* that is the name of another alias group must not contain any upper-case letters.

Lines beginning with white space are treated as continuation lines for the preceding alias. Lines beginning with # are comments.

#### Special Aliases

An alias of the form:

**owner-aliasname : address**

directs error-messages resulting from mail to *aliasname* to *address*, instead of back to the person who sent the message.

An alias of the form:

**aliasname: :include:pathname**

with colons as shown, adds the recipients listed in the file *pathname* to the *aliasname* alias. This allows a private list to be maintained separately from the aliases file.

#### NIS/NIS+ Domain Aliases

The aliases file on the master NIS server is used for the *mail.aliases* NIS map, which can be made available to every NIS client. The *mail\_aliases* table serves the same purpose on a NIS+ server. Thus, the **/etc/mail/aliases\*** files on the various hosts in a network will one day be obsolete. Domain-wide aliases should ultimately be resolved into usernames on specific hosts. For example, if the following were in the domain-wide alias file:

**jsmith:js@jsmachine**

then any NIS/NIS+ client could just mail to **jsmith** and not have to remember the machine and username for John Smith. If a NIS/NIS+ alias does not resolve to an address with a specific host, then the name of the NIS/NIS+ domain is used. There should be an alias of the domain name for a host in this case.

For example, the alias:

```
jsmith:root
```

sends mail on a NIS/NIS+ client to **root@podunk-u** if the name of the NIS/NIS+ domain is **podunk-u**.

#### Automatic Forwarding

When an alias (or address) is resolved to the name of a user on the local host, **sendmail(1M)** checks for a **~/forward** file, owned by the intended recipient, in that user's home directory, and with universal read access. This file can contain one or more addresses or aliases as described above, each of which is sent a copy of the user's mail.

Care must be taken to avoid creating addressing loops in the **~/forward** file. When forwarding mail between machines, be sure that the destination machine does not return the mail to the sender through the operation of any NIS aliases. Otherwise, copies of the message may "bounce." Usually, the solution is to change the NIS alias to direct mail to the proper destination.

A backslash before a username inhibits further aliasing. For instance, to invoke the **vacation** program, user **js** creates a **~/forward** file that contains the line:

```
\js, "|/usr/ucb/vacation js"
```

so that one copy of the message is sent to the user, and another is piped into the **vacation** program.

#### FILES

<b>/etc/passwd</b>	password file
<b>/etc/nisswitch.conf</b>	workstation server definition
<b>/etc/mail/aliases</b>	workstation aliases
<b>/etc/mail/sendmail.cf</b>	sendmail configuration file
<b>~/forward</b>	forwarding information file

#### ATTRIBUTES

See **attributes(5)** for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsr

#### SEE ALSO

**newaliases(1)**, **passwd(1)**, **uucp(1C)**, **vacation(1)**, **sendmail(1M)**, **dbm(3B)**, **passwd(4)**, **attributes(5)**

#### NOTES

Because of restrictions in **dbm(3B)**, a single alias cannot contain more than about 1000 characters. Nested aliases can be used to circumvent this limit.

**NAME** a.out – Executable and Linking Format (ELF) files

**SYNOPSIS** `#include <elf.h>`

**DESCRIPTION**

The file name **a.out** is the default output file name from the link editor, **ld(1)**. The link editor will make an **a.out** executable if there were no errors in linking. The output file of the assembler, **as(1)**, also follows the format of the **a.out** file although its default file name is different.

Programs that manipulate ELF files may use the library that **elf(3E)** describes. An overview of the file format follows. For more complete information, see the references given below.

Linking View	Execution View
ELF header	ELF header
Program header table <i>optional</i>	Program header table
Section 1	Segment 1
...	
Section <i>n</i>	Segment 2
...	...
Section header table	Section header table <i>optional</i>

An ELF header resides at the beginning and holds a “road map” describing the file’s organization. Sections hold the bulk of object file information for the linking view: instructions, data, symbol table, relocation information, and so on. Segments hold the object file information for the program execution view. As shown, a segment may contain one or more sections.

A program header table, if present, tells the system how to create a process image. Files used to build a process image (execute a program) must have a program header table; relocatable files do not need one. A section header table contains information describing the file’s sections. Every section has an entry in the table; each entry gives information such as the section name, the section size, etc. Files used during linking must have a section header table; other object files may or may not have one.

Although the figure shows the program header table immediately after the ELF header, and the section header table following the sections, actual files may differ. Moreover, sections and segments have no specified order. Only the ELF header has a fixed position in the file.

When an **a.out** file is loaded into memory for execution, three logical segments are set up: the text segment, the data segment (initialized data followed by uninitialized, the latter actually being initialized to all 0’s), and a stack. The text segment is not writable by the program; if other processes are executing the same **a.out** file, the processes will share a single text segment.



The data segment starts at the next maximal page boundary past the last text address. If the system supports more than one page size, the “maximal page” is the largest supported size. When the process image is created, the part of the file holding the end of text and the beginning of data may appear twice. The duplicated chunk of text that appears at the beginning of data is never executed; it is duplicated so that the operating system may bring in pieces of the file in multiples of the actual page size without having to realign the beginning of the data section to a page boundary. Therefore, the first data address is the sum of the next maximal page boundary past the end of text plus the remainder of the last text address divided by the maximal page size. If the last text address is a multiple of the maximal page size, no duplication is necessary. The stack is automatically extended as required. The data segment is extended as requested by the **brk(2)** system call.

**SEE ALSO**

**as(1)**, **cc(1B)**, **ld(1)**, **brk(2)**, **elf(3E)**

*ANSI C Programmer's Guide*

<b>NAME</b>	ar – archive file format
<b>SYNOPSIS</b>	<b>#include &lt;ar.h&gt;</b>
<b>DESCRIPTION</b>	<p>The archive command <b>ar</b> is used to combine several files into one. Archives are used mainly as libraries to be searched by the link editor <b>ld</b>.</p> <p>Each archive begins with the archive magic string.</p> <pre>                 #define ARMAG "!&lt;arch&gt;\n"      /* magic string */                 #define SARMAG 8                /* length of magic string */ </pre> <p>Following the archive magic string are the archive file members. Each file member is preceded by a file member header which is of the following format:</p> <pre> #define ARFMAG "`\n"      /* header trailer string */  struct ar_hdr            /* file member header */ {     char  ar_name[16];    /* '/' terminated file member name */     char  ar_date[12];    /* file member date */     char  ar_uid[6];      /* file member user identification */     char  ar_gid[6];      /* file member group identification */     char  ar_mode[8];     /* file member mode (octal) */     char  ar_size[10];    /* file member size */     char  ar_fmags[2];    /* header trailer string */ }; </pre> <p>All information in the file member headers is in printable ASCII. The numeric information contained in the headers is stored as decimal numbers (except for <i>ar_mode</i> which is in octal). Thus, if the archive contains printable files, the archive itself is printable.</p> <p>If the file member name fits, the <i>ar_name</i> field contains the name directly, and is terminated by a slash (/) and padded with blanks on the right. If the member's name does not fit, <i>ar_name</i> contains a slash (/) followed by a decimal representation of the name's offset in the archive string table described below.</p> <p>The <i>ar_date</i> field is the modification date of the file at the time of its insertion into the archive. Common format archives can be moved from system to system as long as the portable archive command <b>ar</b> is used.</p> <p>Each archive file member begins on an even byte boundary; a newline is inserted between files if necessary. Nevertheless, the size given reflects the actual size of the file exclusive of padding.</p> <p>Notice there is no provision for empty areas in an archive file.</p>

Each archive that contains object files (see **a.out(4)**) includes an archive symbol table. This symbol table is used by the link editor **ld** to determine which archive members must be loaded during the link edit process. The archive symbol table (if it exists) is always the first file in the archive (but is never listed) and is automatically created and/or updated by **ar**.

The archive symbol table has a zero length name (that is, **ar\_name[0]** is **'**), **ar\_name[1]** is **'**, etc.). All “words” in this symbol table have four bytes, using the machine-independent encoding shown below. All machines use the encoding described here for the symbol table, even if the machine’s “natural” byte order is different.

<b>0x01020304</b>	0	1	2	3
	01	02	03	04

The contents of this file are as follows:

1. The number of symbols. Length: 4 bytes.
2. The array of offsets into the archive file. Length: 4 bytes \* “the number of symbols”.
3. The name string table. Length: *ar\_size* – 4 bytes \* (“the number of symbols” + 1).

As an example, the following symbol table defines 4 symbols. The archive member at file offset 114 defines *name* and *object*. The archive member at file offset 426 defines *function* and a second version of *name*.

**Example Symbol Table**

Offset	+0	+1	+2	+3		
0	4				4 offset entries	
4	114					<b>name</b>
8	114					<b>object</b>
12	426					<b>function</b>
16	426				<b>name</b>	
20	n	a	m	e		
24	\0	o	b	j		
28	e	c	t	\0		
32	f	u	n	c		
36	t	i	o	n		
40	\0	n	a	m		
44	e	\0				

The string table contains exactly as many null terminated strings as there are elements in the offsets array. Each offset from the array is associated with the corresponding name from the string table (in order). The names in the string table are all the defined global symbols found in the common object files in the archive. Each offset is the location of the archive header for the associated symbol.

If some archive member's name is more than 15 bytes long, a special archive member contains a table of file names, each followed by a slash and a new-line. This string table member, if present, will precede all "normal" archive members. The special archive symbol table is not a "normal" member, and must be first if it exists. The **ar\_name** entry of the string table's member header holds a zero length name **ar\_name[0]**=='/', followed by one trailing slash (**ar\_name[1]**=='/'), followed by blanks (**ar\_name[2]**==' ', etc.). Offsets into the string table begin at zero. Example *ar\_name* values for short and long file names appear below.

Offset	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
0	f	i	l	e	_	n	a	m	e	_
10	s	a	m	p	l	e	/	\n	l	o
20	n	g	e	r	f	i	l	e	n	a
30	m	e	x	a	m	p	l	e	/	\n

Member Name	ar_name
<b>short-name</b>	Not in string table
<b>file_name_sample</b>	Offset 0 in string table
<b>longerfilenameexample</b>	Offset 18 in string table

**SEE ALSO** ar(1), ld(1), strip(1), a.out(4)

**NOTES** **strip** will remove all archive symbol entries from the header. The archive symbol entries must be restored via the **-ts** options of the **ar** command before the archive can be used with the link editor **ld**.

NAME	DESCRIPTION
archives – device header	<pre> /* Magic numbers */ #define CMN_ASC 0x070701 /* Cpio Magic Number for -c header */ #define CMN_BIN 070707 /* Cpio Magic Number for Binary header */ #define CMN_BBS 0143561 /* Cpio Magic Number for Byte-Swap header */ #define CMN_CRC 0x070702 /* Cpio Magic Number for CRC header */ #define CMS_ASC "070701" /* Cpio Magic String for -c header */ #define CMS_CHR "070707" /* Cpio Magic String for odc header */ #define CMS_CRC "070702" /* Cpio Magic String for CRC header */ #define CMS_LEN 6 /* Cpio Magic String length */  /* Various header and field lengths */ #define CHRSZ 76 /* -H odc size minus filename field */ #define ASCSZ 110 /* -c and CRC hdr size minus filename field */ #define TARSZ 512 /* TAR hdr size */  #define HNAMLN 256 /* maximum filename length for binary and odc headers */ #define EXPNLN 1024 /* maximum filename length for -c and CRC headers */ #define HTIMLEN 2 /* length of modification time field */ #define HSIZLEN 2 /* length of file size field */  /* cpio binary header definition */ struct hdr_cpio {     short h_magic, /* magic number field */           h_dev; /* file system of file */     ushort h_ino, /* inode of file */            h_mode, /* modes of file */            h_uid, /* uid of file */            h_gid; /* gid of file */     short h_nlink, /* number of links to file */           h_rdev, /* maj/min numbers for special files */           h_mtime[HTIMLEN], /* modification time of file */           h_namesize, /* length of filename */           h_filesize[HSIZLEN]; /* size of file */     char h_name[HNAMLN]; /* filename */ };  /* cpio -H odc header format */ struct c_hdr {     char c_magic[CMS_LEN],           c_dev[6],           c_ino[6],           c_mode[6],           c_uid[6],           c_gid[6],           c_nlink[6], </pre>

```

        c_rdev[6],
        c_mtime[11],
        c_namesz[6],
        c_filesz[11],
        c_name[HNAMLEN];
};
/* -c and CRC header format */
struct Exp_cpio_hdr {
    char    E_magic[CMS_LEN],
           E_ino[8],
           E_mode[8],
           E_uid[8],
           E_gid[8],
           E_nlink[8],
           E_mtime[8],
           E_filesize[8],
           E_maj[8],
           E_min[8],
           E_rmaj[8],
           E_rmin[8],
           E_namesize[8],
           E_chksm[8],
           E_name[EXP_NLEN];
};
/* Tar header structure and format */
#define TBLOCK    512      /* length of tar header and data blocks */
#define TNAMLEN   100     /* maximum length for tar file names */
#define TMODLEN   8       /* length of mode field */
#define TUIDLEN   8       /* length of uid field */
#define TGIDLEN   8       /* length of gid field */
#define TSIZLEN   12      /* length of size field */
#define TTIMLEN   12      /* length of modification time field */
#define TCRCLLEN  8       /* length of header checksum field */
/* tar header definition */
union tblock {
    char dummy[TBLOCK];
    struct header {
        char t_name[TNAMLEN];          /* name of file */
        char t_mode[TMODLEN];          /* mode of file */
        char t_uid[TUIDLEN];           /* uid of file */
        char t_gid[TGIDLEN];           /* gid of file */
        char t_size[TSIZLEN];          /* size of file in bytes */
        char t_mtime[TTIMLEN];         /* modification time of file */
        char t_chksm[TCRCLLEN];        /* checksum of header */
        char t_typeflag;                /* flag to indicate type of file */
        char t_linkname[TNAMLEN];      /* file this file is linked with */
        char t_magic[6];                /* magic string always "ustar" */
    };
};

```

```

        char t_version[2];           /* version strings always "00" */
        char t_uname[32];           /* owner of file in ASCII */
        char t_gname[32];           /* group of file in ASCII */
        char t_devmajor[8];         /* major number for special files */
        char t_devminor[8];        /* minor number for special files */
        char t_prefix[155];         /* pathname prefix */
    } tbuf;
};
/* volcopy tape label format and structure */
#define VMAGLEN8
#define VVOLLEN6
#define VFILLEN 464
struct volcopy_label {
    char v_magic[VMAGLEN],
        v_volume[VVOLLEN],
        v_reels,
        v_reel;
    long v_time,
        v_length,
        v_dens,
        v_reelblks, /* u370 added field */
        v_blksize, /* u370 added field */
        v_nblocks; /* u370 added field */
    char v_fill[VFILLEN];
    long v_offset; /* used with -e and -reel options */
    int v_type; /* does tape have nblocks field? */
};

```

<b>NAME</b>	asetenv – ASET environment file														
<b>SYNOPSIS</b>	<code>/usr/aset/asetenv</code>														
<b>DESCRIPTION</b>	<p>The <code>asetenv</code> file is located in <code>/usr/aset</code>, the default operating directory of the Automated Security Enhancement Tool (ASET). An alternative working directory can be specified by the administrators through the <code>aset -d</code> command or the <code>ASETDIR</code> environment variable. See <code>aset(1M)</code>. <code>asetenv</code> contains definitions of environment variables for ASET.</p> <p>There are 2 sections in this file. The first section is labeled <i>User Configurable Parameters</i>. It contains, as the label indicates, environment variables that the administrators can modify to customize ASET behavior to suit their specific needs. The second section is labeled <i>ASET Internal Environment Variables</i> and should not be changed. The configurable parameters are explained as follows:</p> <p><b>TASK</b> This variable defines the list of tasks that <code>aset</code> will execute the next time it runs. The available tasks are:</p> <table border="0"> <tr> <td style="padding-right: 20px;"><b>tune</b></td> <td>Tighten system files.</td> </tr> <tr> <td><b>usrgrp</b></td> <td>Check user/group.</td> </tr> <tr> <td><b>sysconf</b></td> <td>Check system configuration file.</td> </tr> <tr> <td><b>env</b></td> <td>Check environment.</td> </tr> <tr> <td><b>cklist</b></td> <td>Compare system files checklist.</td> </tr> <tr> <td><b>eeprom</b></td> <td>Check <code>eeprom(1M)</code> parameters.</td> </tr> <tr> <td><b>firewall</b></td> <td>Disable forwarding of IP packets.</td> </tr> </table> <p><b>CKLISTPATH_LOW</b>  <b>CKLISTPATH_MED</b>  <b>CKLISTPATH_HIGH</b></p> <p>These variables define the list of directories to be used by <code>aset</code> to create a <i>checklist</i> file at the <i>low</i>, <i>medium</i>, and <i>high</i> security levels, respectively. Attributes of all the files in the directories defined by these variables will be checked periodically and any changes will be reported by <code>aset</code>. Checks performed on these directories are not recursive. <code>aset</code> only checks directories explicitly listed in these variables and does not check subdirectories of them.</p> <p><b>YPCHECK</b></p> <p>This variable is a boolean parameter. It specifies whether <code>aset</code> should extend checking (when applicable) on system tables to their NIS equivalents or not. The value <b>true</b> enables it while the value <b>false</b> disables it.</p>	<b>tune</b>	Tighten system files.	<b>usrgrp</b>	Check user/group.	<b>sysconf</b>	Check system configuration file.	<b>env</b>	Check environment.	<b>cklist</b>	Compare system files checklist.	<b>eeprom</b>	Check <code>eeprom(1M)</code> parameters.	<b>firewall</b>	Disable forwarding of IP packets.
<b>tune</b>	Tighten system files.														
<b>usrgrp</b>	Check user/group.														
<b>sysconf</b>	Check system configuration file.														
<b>env</b>	Check environment.														
<b>cklist</b>	Compare system files checklist.														
<b>eeprom</b>	Check <code>eeprom(1M)</code> parameters.														
<b>firewall</b>	Disable forwarding of IP packets.														



**UID\_ALIASES**

This variable specifies an alias file for user ID sharing. Normally, **aset** warns about multiple user accounts sharing the same user ID because it is not advisable for accountability reason. Exceptions can be created using an alias file. User ID sharing allowed by the alias file will not be reported by **aset**. See **asetmasters(4)** for the format of the alias file.

**PERIODIC\_SCHEDULE**

This variable specifies the schedule for periodic execution of ASET. It uses the format of **crontab(1)** entries. Briefly speaking, the variable is assigned a string of the following format:

*minutes hours day-of-month month day-of-week*

Setting this variable does *not* activate the periodic schedule of ASET. To execute ASET periodically, **aset(1M)** must be run with the **-p** option. See **aset(1M)**. For example, if **PERIODIC\_SCHEDULE** is set to the following, and **aset(1M)** was started with the **-p** option, **aset** will run at 12:00 midnight every day:

**00 \* \* \***

**EXAMPLES**

The following is a sample **asetenv** file, showing the settings of the ASET configurable parameters:

```
CKLISTPATH_LOW=/etc/
CKLISTPATH_MED=$CHECKLISTPATH_LOW:/usr/bin:/usr/ucb
CKLISTPATH_HIGH=$CHECKLISTPATH_MED:/usr/lib:/usr/sbin
YPCHECK=false
UID_ALIASES=/usr/aset/masters/uid_aliases
PERIODIC_SCHEDULE="00 * * *"
TASKS="env sysconf usrgrp"
```

When **aset -p** is run with this file, **aset** is executed at midnight of every day. The **/** and **/etc** directories are checked at the *low* security level; the **/**, **/etc**, **/usr/bin**, and **/usr/ucb** directories are checked at the *medium* security level; and the **/**, **/etc**, **/usr/bin**, **/usr/lib**, and **/usr/sbin** directories are checked at the *high* security level. Checking of NIS system files is disabled. The **/usr/aset/masters/uid\_aliases** file specifies the used IDs available for sharing. The **env**, **sysconf**, and **usrgrp** tasks will be performed, checking the environment variables, various system tables, and the local **passwd** and **group** files.

**SEE ALSO**

**crontab(1)**, **aset(1M)**, **asetmasters(4)**  
*ASET Administrator Manual*

<b>NAME</b>	asetmasters, tune.low, tune.med, tune.high, uid_aliases, cklist.low, cklist.med, cklist.high – ASET master files										
<b>SYNOPSIS</b>	<b>/usr/aset/masters/tune.low</b> <b>/usr/aset/masters/tune.med</b> <b>/usr/aset/masters/tune.high</b> <b>/usr/aset/masters/uid_aliases</b> <b>/usr/aset/masters/cklist.low</b> <b>/usr/aset/masters/cklist.med</b> <b>/usr/aset/masters/cklist.high</b>										
<b>DESCRIPTION</b>	<p>The <b>/usr/aset/masters</b> directory contains several files used by the Automated Security Enhancement Tool (ASET). <b>/usr/aset</b> is the default operating directory for ASET. An alternative working directory can be specified by the administrators through the <b>aset -d</b> command or the <b>ASETDIR</b> environment variable. See <b>aset(1M)</b>.</p> <p>These files are provided by default to meet the need of most environments. The administrators, however, can edit these files to meet their specific needs. The format and usage of these files are described below.</p> <p>All the master files allow comments and blank lines to improve readability. Comment lines must start with a leading "#" character.</p> <p><b>tune.low</b> <b>tune.med</b> <b>tune.high</b>    These files are used by the <b>tune</b> task (see <b>aset(1M)</b>) to restrict the permission settings for system objects. Each file is used by ASET at the security level indicated by the suffix. Each entry in the files is of the form:</p> <p style="padding-left: 2em;"><i>pathname mode owner group type</i></p> <p>where</p> <table border="0" style="margin-left: 2em;"> <tr> <td><i>pathname</i></td> <td>is the full pathname</td> </tr> <tr> <td><i>mode</i></td> <td>is the permission setting</td> </tr> <tr> <td><i>owner</i></td> <td>is the owner of the object</td> </tr> <tr> <td><i>group</i></td> <td>is the group of the object</td> </tr> <tr> <td><i>type</i></td> <td>is the type of the object It can be <b>symlink</b> for a symbolic link, <b>directory</b> for a directory, or <b>file</b> for everything else.</td> </tr> </table> <p>Regular shell wildcard ("*", "?", ...) characters can be used in the <i>pathname</i> for multiple references. See <b>sh(1)</b>. The <i>mode</i> is a five-digit number that represents the permission setting. Note that this setting represents a least restrictive value. If the current setting is already more restrictive than the specified value, ASET does not loosen the permission settings.</p>	<i>pathname</i>	is the full pathname	<i>mode</i>	is the permission setting	<i>owner</i>	is the owner of the object	<i>group</i>	is the group of the object	<i>type</i>	is the type of the object It can be <b>symlink</b> for a symbolic link, <b>directory</b> for a directory, or <b>file</b> for everything else.
<i>pathname</i>	is the full pathname										
<i>mode</i>	is the permission setting										
<i>owner</i>	is the owner of the object										
<i>group</i>	is the group of the object										
<i>type</i>	is the type of the object It can be <b>symlink</b> for a symbolic link, <b>directory</b> for a directory, or <b>file</b> for everything else.										

For example, if *mode* is **00777**, the permission will not be changed, since it is always less restrictive than the current setting.

Names must be used for *owner* and *group* instead of numeric ID's. *?* can be used as a "don't care" character in place of *owner*, *group*, and *type* to prevent ASET from changing the existing values of these parameters.

#### **uid\_alias**

This file allows user ID's to be shared by multiple user accounts. Normally, ASET discourages such sharing for accountability reason and reports user ID's that are shared. The administrators can, however, define permissible sharing by adding entries to the file. Each entry is of the form:

```
uid=alias1=alias2=alias3= ...
```

where

*uid* is the shared user id

*alias?* is the user accounts sharing the user ID

For example, if **sync** and **daemon** share the user ID **1**, the corresponding entry is:

```
1=sync=daemon
```

#### **cklist.low**

#### **cklist.med**

#### **cklist.high**

These files are used by the **cklist** task (see **aset(1M)**), and are created the first time the task is run at the *low*, *medium*, and *high* levels. When the **cklist** task is run, it compares the specified directory's contents with the appropriate **cklist.level** file and reports any discrepancies.

#### **EXAMPLES**

The following is an example of valid entries for the **tune.low**, **tune.med**, and **tune.high** files:

```
/bin      00777    root      staff     symlink
/etc      02755    root      staff     directory
/dev/sd*  00640    root      operator  file
```

#### **SEE ALSO**

**aset(1M)**, **asetenv(4)**

*ASET Administrator Manual*

<b>NAME</b>	audit_class – audit class definitions
<b>SYNOPSIS</b>	<i>/etc/security/audit_class</i>
<b>DESCRIPTION</b>	<p><i>/etc/security/audit_class</i> is an ASCII system file that stores class definitions. Programs use the <b>getauclassent(3)</b> routines to access this information.</p> <p>The fields for each class entry are separated by colons. Each class entry is a bitmap and is separated from each other by a newline.</p> <p>Each entry in the <i>audit_class</i> file has the form:</p> <p style="padding-left: 40px;"><i>mask:name:description</i></p> <p>The fields are defined as follows:</p> <p><i>mask</i>                    The class mask.</p> <p><i>name</i>                    The class name.</p> <p><i>description</i>            The description of the class.</p> <p>The classes are now user-configurable. Each class is represented as a bit in the class mask which is an unsigned integer. Thus, there are 32 different classes available, plus two meta-classes -- <b>all</b> and <b>no</b>.</p> <p style="padding-left: 40px;"><b>all</b> represents a conjunction of all allowed classes, and is provided as a shorthand method of specifying all classes.</p> <p style="padding-left: 40px;"><b>no</b> is the "invalid" class, and any event mapped solely to this class will not be audited. (Turning auditing on to the <b>all</b> meta class will NOT cause events mapped solely to the <b>no</b> class to be written to the audit trail.)</p>
<b>EXAMPLES</b>	<p>Here is a sample of an <i>audit_class</i> file:</p> <pre style="padding-left: 40px;">0x00000000:no:invalid class 0x00000001:fr:file read 0x00000002:fw:file write 0x00000004:fa:file attribute access 0x00000008:fm:file attribute modify 0x00000010:fc:file create 0x00000020:fd:file delete 0x00000040:cl:file close 0xffffffff:all:all classes</pre>
<b>FILES</b>	<i>/etc/security/audit_class</i>
<b>SEE ALSO</b>	<b>bsmconv(1M)</b> , <b>getauclassent(3)</b> , <b>audit_event(4)</b>
<b>NOTES</b>	It is possible to deliberately turn on the <b>no</b> class in the kernel, in which case the audit trail will be flooded with records for the audit event AUE_NULL.

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.

<b>NAME</b>	audit_control – control information for system audit daemon
<b>SYNOPSIS</b>	<b>/etc/security/audit_control</b>
<b>DESCRIPTION</b>	<p>The <b>audit_control</b> file contains audit control information used by <b>auditd(1M)</b>. Each line consists of a title and a string, separated by a colon. There are no restrictions on the order of lines in the file, although some lines must appear only once. A line beginning with '#' is a comment.</p> <p>Directory definition lines list the directories to be used when creating audit files, in the order in which they are to be used. The format of a directory line is:</p> <p style="padding-left: 40px;"><b>dir:</b> <i>directory-name</i></p> <p><i>directory-name</i> is where the audit files will be created. Any valid writable directory can be specified.</p> <p>The following configuration is recommended:</p> <p style="padding-left: 40px;"><b>/etc/security/audit/server/files</b></p> <p>where <i>server</i> is the name of a central machine, since audit files belonging to different servers are usually stored in separate subdirectories of a single audit directory. The naming convention normally has <i>server</i> be a directory on a server machine, and all clients mount <b>/etc/security/audit/server</b> at the same location in their local file systems. If the same server exports several different file systems for auditing, their <i>server</i> names will, of course, be different.</p> <p>There are several other ways for audit data to be arranged: some sites may have needs more in line with storing each host's audit data in separate subdirectories. The audit structure used will depend on each individual site.</p> <p>The audit threshold line specifies the percentage of free space that must be present in the file system containing the current audit file. The format of the threshold line is:</p> <p style="padding-left: 40px;"><b>minfree:</b> <i>percentage</i></p> <p>where <i>percentage</i> indicates the amount of free space required. If free space falls below this threshold, the audit daemon <b>auditd(1M)</b> invokes the shell script <b>audit_warn(1M)</b>. If no threshold is specified, the default is 0%.</p> <p>The audit flags line specifies the default system audit value. This value is combined with the user audit value read from <b>audit_user(4)</b> to form the process audit state. The user audit value overrides the system audit value. The format of a flags line is:</p> <p style="padding-left: 40px;"><b>flags:</b><i>audit-flags</i></p>

where *audit-flags* specifies which event classes are to be audited. The character string representation of *audit-flags* contains a series of flag names, each one identifying a single audit class, separated by commas. A name preceded by '-' means that the class should be audited for failure only; successful attempts are not audited. A name preceded by '+' means that the class should be audited for success only; failing attempts are not audited. Without a prefix, the name indicates that the class is to be audited for both successes and failures. The special string **all** indicates that all events should be audited; **-all** indicates that all failed attempts are to be audited, and **+all** all successful attempts. The prefixes ^, ^-, and ^+ turn off flags specified earlier in the string (^- and ^+ for failing and successful attempts, ^ for both). They are typically used to reset flags.

The non-attributable flags line is similar to the flags line, but this one contain the audit flags that define what classes of events are audited when an action cannot be attributed to a specific user. The format of a **naflags** line is:

**naflags:** *audit-flags*

The flags are separated by commas, with no spaces.

The following table lists the predefined audit classes:

short name	long name	short description
no	no_class	null value for turning off event preselection
fr	file_read	Read of data, open for reading, etc.
fw	file_write	Write of data, open for writing, etc.
fa	file_attr_acc	Access of object attributes: stat, pathconf, etc.
fm	file_attr_mod	Change of object attributes: chown, flock, etc.
fc	file_creation	Creation of object
fd	file_deletion	Deletion of object
cl	file_close	close(2) system call
pc	process	Process operations: fork, exec, exit, etc.
nt	network	Network events: bind, connect, accept, etc.
ip	ipc	System V IPC operations
na	non_attrib	non-attributable events
ad	administrative	administrative actions: mount, exportfs, etc.
lo	login_logout	Login and logout events
ap	application	Application auditing
io	ioctl	ioctl(2) system call
ex	exec	exec(2) system call
ot	other	Everything else
all	all	All flags set

Note that the classes are configurable, see **audit\_class(4)**.

**EXAMPLES**

Here is a sample `/etc/security/audit_control` file for the machine `eggplant`:

```

dir: /etc/security/jedgar/eggplant
dir: /etc/security/jedgar.aux/eggplant
#
# Last-ditch audit file system when jedgar fills up.
#
dir: /etc/security/global/eggplant
minfree: 20
flags: lo,ad,-all,^-fm
naflags: lo,ad

```

This identifies server **jedgar** with two file systems normally used for audit data, another server **global** used only when **jedgar** fills up or breaks, and specifies that the warning script is run when the file systems are 80% filled. It also specifies that all logins, administrative operations are to be audited (whether or not they succeed), and that failures of all types except failures to access object attributes are to be audited.

**FILES**

```

/etc/security/audit_control
/etc/security/audit_warn
/etc/security/audit/**/*
/etc/security/audit_user

```

**SEE ALSO**

**audit(1M)**, **audit\_warn(1M)**, **auditd(1M)**, **bsmconv(1M)**, **audit(2)**, **getfauditflags(3)**, **audit.log(4)**, **audit\_class(4)**, **audit\_user(4)**

**NOTES**

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.



<b>NAME</b>	audit_data – current information on audit daemon
<b>SYNOPSIS</b>	<b>/etc/security/audit_data</b>
<b>DESCRIPTION</b>	<p>The <b>audit_data</b> file contains information about the audit daemon. The file contains the process ID of the audit daemon, and the pathname of the current audit log file. The format of the file is:</p> <p style="text-align: center;"><i>&lt;pid&gt;:&lt;pathname&gt;</i></p> <p>Where <i>pid</i> is the process ID for the audit daemon, and <i>pathname</i> is the full pathname for the current audit log file.</p>
<b>EXAMPLES</b>	<b>64:/etc/security/audit/server1/19930506081249.19930506230945.bongos</b>
<b>FILES</b>	<b>/etc/security/audit_data</b>
<b>SEE ALSO</b>	<b>audit(1M), auditd(1M), bsmconv(1M), audit(2), audit.log(4)</b>
<b>NOTES</b>	The functionality described in this man page is available only if the Basic Security Module (BSM) has been enabled. See <b>bsmconv(1M)</b> for more information.

<b>NAME</b>	audit_event – audit event definition and class mapping
<b>SYNOPSIS</b>	<i>/etc/security/audit_event</i>
<b>DESCRIPTION</b>	<p><i>/etc/security/audit_event</i> is an ASCII system file that stores event definitions and specifies the event to class mappings. Programs use the <b>getauevent(3)</b> routines to access this information.</p> <p>The fields for each event entry are separated by colons. Each event is separated from the next by a newline.</p> <p>Each entry in the <i>audit_event</i> file has the form:</p> <p style="padding-left: 40px;"><i>number:name:description:flags</i></p> <p>The fields are defined as follows:</p> <p><i>number</i>                    The event number.</p> <p><i>name</i>                        The event name.</p> <p><i>description</i>                The description of the event.</p> <p><i>flags</i>                        Flags specifying classes to which the event is mapped.</p>
<b>EXAMPLES</b>	<p>Here is a sample of the <i>audit_event</i> file entries:</p> <p style="padding-left: 40px;"><b>7:AUE_EXEC:exec(2):pc.ex</b></p> <p style="padding-left: 40px;"><b>79:AUE_OPEN_WTC:open(2) - write,creat,trunc:fc,fd,fw</b></p> <p style="padding-left: 40px;"><b>6152:AUE_login:login - success or failure:lo</b></p> <p style="padding-left: 40px;"><b>6153:AUE_logout:logout:lo</b></p> <p style="padding-left: 40px;"><b>6154:AUE_telnet:login - through telnet:lo</b></p> <p style="padding-left: 40px;"><b>6155:AUE_rlogin:login - through rlogin:lo</b></p>
<b>FILES</b>	<i>/etc/security/audit_event</i>
<b>SEE ALSO</b>	<b>bsmconv(1M)</b> , <b>getauevent(3)</b> , <b>audit_control(4)</b>
<b>NOTES</b>	The functionality described in this man page is available only if the Basic Security Module (BSM) has been enabled. See <b>bsmconv(1M)</b> for more information.

<b>NAME</b>	audit.log – audit trail file																														
<b>SYNOPSIS</b>	<b>#include</b> <bsm/audit.h> <b>#include</b> <bsm/audit_record.h>																														
<b>DESCRIPTION</b>	<p><b>audit.log</b> files are the depository for audit records stored locally or on an audit server. These files are kept in directories named in the file <b>audit_control(4)</b>. They are named to reflect the time they are created and are, when possible, renamed to reflect the time they are closed as well. The name takes the form</p> <p style="text-align: center;"><i>yyyymmddhhmss.not_terminated.hostname</i></p> <p>when open or if the <b>auditd(1M)</b> terminated ungracefully, and the form</p> <p style="text-align: center;"><i>yyyymmddhhmss.yyyyymmddhhmss.hostname</i></p> <p>when properly closed. <b>yyyy</b> is the year, <b>mm</b> the month, <b>dd</b> day in the month, <b>hh</b> hour in the day, <b>mm</b> minute in the hour, and <b>ss</b> second in the minute. All fields are of fixed width.</p> <p>The <b>audit.log</b> file begins with a standalone <b>file token</b> and typically ends with one also. The beginning <b>file token</b> records the pathname of the previous audit file, while the ending <b>file token</b> records the pathname of the next audit file. If the file name is NULL the appropriate path was unavailable.</p> <p>The <b>audit.log</b> files contains audit records. Each audit record is made up of <i>audit tokens</i>. Each record contains a header token followed by various data tokens. Depending on the audit policy in place by <b>auditon(2)</b>, optional other tokens such as trailers or sequences may be included.</p> <p>The tokens are defined as follows:</p> <p>The <b>file</b> token consists of:</p> <table border="0" style="margin-left: 20px;"> <tr><td>token ID</td><td>char</td></tr> <tr><td>seconds of time</td><td>u_int</td></tr> <tr><td>milliseconds of time</td><td>u_int</td></tr> <tr><td>file name length</td><td>short</td></tr> <tr><td>file pathname</td><td>null terminated string</td></tr> </table> <p>The <b>header</b> token consists of:</p> <table border="0" style="margin-left: 20px;"> <tr><td>token ID</td><td>char</td></tr> <tr><td>record byte count</td><td>u_long</td></tr> <tr><td>version #</td><td>char (1)</td></tr> <tr><td>event type</td><td>u_short</td></tr> <tr><td>event modifier</td><td>u_short</td></tr> <tr><td>seconds of time</td><td>u_int</td></tr> <tr><td>milliseconds of time</td><td>u_int</td></tr> </table> <p>The <b>trailer</b> token consists of:</p> <table border="0" style="margin-left: 20px;"> <tr><td>token ID</td><td>char</td></tr> <tr><td>trailer magic number</td><td>u_short</td></tr> <tr><td>record byte count</td><td>u_long</td></tr> </table>	token ID	char	seconds of time	u_int	milliseconds of time	u_int	file name length	short	file pathname	null terminated string	token ID	char	record byte count	u_long	version #	char (1)	event type	u_short	event modifier	u_short	seconds of time	u_int	milliseconds of time	u_int	token ID	char	trailer magic number	u_short	record byte count	u_long
token ID	char																														
seconds of time	u_int																														
milliseconds of time	u_int																														
file name length	short																														
file pathname	null terminated string																														
token ID	char																														
record byte count	u_long																														
version #	char (1)																														
event type	u_short																														
event modifier	u_short																														
seconds of time	u_int																														
milliseconds of time	u_int																														
token ID	char																														
trailer magic number	u_short																														
record byte count	u_long																														

The <b>arbitrary data</b> token is defined:	
token ID	char
how to print	char
basic unit	char
unit count	char
data items	<i>depends on basic unit</i>
The <b>in_addr</b> token consists of:	
token ID	char
internet address	char
The <b>ip</b> token consists of:	
token ID	char
version and ihl	char
type of service	char
length	short
id	u_short
offset	u_short
ttl	char
protocol	char
checksum	u_short
source address	long
destination address	long
The <b>ipport</b> token consists of:	
token ID	char
port address	short
The <b>opaque</b> token consists of:	
token ID	char
size	short
data	char, <i>size</i> chars
The <b>path</b> token consists of:	
token ID	char
path length	short
path	null terminated string
The <b>process</b> token consists of:	
token ID	char
auid	u_long
euid	u_long
egid	u_long
ruid	u_long
rgid	u_long
pid	u_long
sid	u_long
terminal ID	u_long (port ID)
	u_long (machine ID)

The <b>return</b> token consists of:	
token ID	char
error number	char
return value	long
The <b>subject</b> token consists of:	
token ID	char
auid	u_long
euid	u_long
egid	u_long
ruid	u_long
rgid	u_long
pid	u_long
sid	u_long
terminal ID	u_long (port ID) u_long (machine ID)
The <b>System V IPC</b> token consists of:	
token ID	char
object ID type	char
object ID	long
The <b>text</b> token consists of:	
token ID	char
text length	short
text	null terminated string
The <b>attribute</b> token consists of:	
token ID	char
mode	u_long
uid	u_long
gid	u_long
file system id	long
node id	long
device	u_long
The <b>groups</b> token consists of:	
token ID	char
number	short
group list	long, <i>size</i> chars
The <b>System V IPC permission</b> token consists of:	
token ID	char
uid	u_long
gid	u_long
cuid	u_long
cgid	u_long
mode	u_long
seq	u_long

key	long
The <b>arg</b> token consists of:	
token ID	char
argument #	char
argument value	long
string length	short
text	null terminated string
The <b>exec_args</b> token consists of:	
token ID	char
count	long
text	<i>count</i> null terminated string(s)
The <b>exec_env</b> token consists of:	
token ID	char
count	long
text	<i>count</i> null terminated string(s)
The <b>exit</b> token consists of:	
token ID	char
status	long
return value	long
The <b>socket</b> token consists of:	
token ID	char
socket type	short
local port	short
local Internet address	char
remote port	short
remote Internet address	char
The <b>seq</b> token consists of:	
token ID	char
sequence number	long

**SEE ALSO** **audit(1M)**, **auditd(1M)**, **bsmconv(1M)**, **audit(2)**, **auditon(2)**, **au\_to(3)**, **audit\_control(4)**

**NOTES**

Each token is generally written using the **au\_to(3)** family of function calls. 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.

<b>NAME</b>	audit_user – per-user auditing data file
<b>SYNOPSIS</b>	<i>/etc/security/audit_user</i>
<b>DESCRIPTION</b>	<p><b>audit_user</b> is an access-restricted ASCII system file that stores per-user auditing preselection data. Programs use the <b>getauusernam</b>(3) routines to access this information.</p> <p>The fields for each user entry are separated by colons. Each user is separated from the next by a newline. <b>audit_user</b> does not have general read permission.</p> <p>Each entry in the audit_user file has the form:</p> <p style="padding-left: 40px;"><i>username:always-audit-flags:never-audit-flags</i></p> <p>The fields are defined as follows:</p> <p><i>username</i>                    The user's login name.</p> <p><i>always-audit-flags</i>        Flags specifying event classes to <i>always</i> audit.</p> <p><i>never-audit-flags</i>        Flags specifying event classes to <i>never</i> audit.</p>
<b>EXAMPLES</b>	<p>Here is a sample <b>audit_user</b> file:</p> <p style="padding-left: 40px;"><b>other:lo,ad:io,cl</b>  <b>fred:lo,ex,+fc,-fr,-fa:io,cl</b>  <b>ethyl:lo,ex,nt:io,cl</b></p>
<b>FILES</b>	<i>/etc/security/audit_user</i> <i>/etc/passwd</i>
<b>SEE ALSO</b>	<b>bsmconv</b> (1M), <b>getauusernam</b> (3), <b>audit_control</b> (4), <b>passwd</b> (4),
<b>NOTES</b>	The functionality described in this man page is available only if the Basic Security Module (BSM) has been enabled. See <b>bsmconv</b> (1M) for more information.

<b>NAME</b>	bootparams – boot parameter data base
<b>SYNOPSIS</b>	<b>/etc/bootparams</b>
<b>DESCRIPTION</b>	<p>The <b>bootparams</b> file contains a list of client entries that diskless clients use for booting. Diskless booting clients retrieve this information by issuing requests to a server running the <b>rpc.bootparamd</b>(1M) program. The <b>bootparams</b> file may be used in conjunction with or in place of other sources for the <b>bootparams</b> information. See <b>nsswitch.conf</b>(4).</p> <p>For each client the file contains an entry with the client's name and a list of boot parameter values for that client. Each entry should have the form:</p> <p style="text-align: center;"><i>clientname identifier-specifier ...</i></p> <p>The first item of each entry is the host name of the diskless client. The asterisk (*) character may be used as a "wildcard" in place of the client name in a single entry. That entry will apply to all clients for whom there is not an entry that specifically names them.</p> <p>This is followed by one or more whitespace characters and a series of identifier-specifiers separated by whitespace characters.</p> <p>Each identifier-specifier has the form:</p> <p style="text-align: center;"><i>identifier=server:pathname</i></p> <p>or</p> <p style="text-align: center;"><i>identifier=domain-name</i></p> <p>The first form is used for file-specific identifiers. A file-specific <i>identifier</i> is a key that is used by diskless clients to identify a file or filesystem. <i>server</i> is the name of the server that will provide the file or filesystem to the diskless client, and <i>pathname</i> is the path to the exported file or filesystem on the specified server. The equal sign (=) and colon (:) characters are used in the indicated positions. There should not be any whitespace within an identifier-specifier.</p> <p>Non-file-specific identifiers use the second form of identifier-specifier. One non-file-specific value for <i>identifier</i> is supported: the assignment of the client's domain name. In this case, the value used for <i>identifier</i> is <b>domain</b>. <i>domain-name</i> must be the client's domain name. The algorithm for determining a client's domain name is to first check for a <b>domain</b> identifier in the client-specific entry and then in "wildcard" entry. If none is found, the server's domain name is used.</p> <p>An entry may be split across multiple lines of the file. The backslash (\) character should be used as the last character of a line to signify that the entry continues on the next line. The line may only be split in places where whitespace is allowed in the entry.</p> <p>A variation of the first form (<i>identifier=server:pathname</i>) is used for the <b>ns</b> key which forces <b>sysidtool</b>(1M) to use a specific name service. By default, <b>sysidtool</b> uses NIS+ in preference to NIS if it can find a NIS+ server for the system's domain on the subnet. This key may be necessary if you are trying to set up a hands-off installation, or if the name server is on a different subnet, which is common with NIS+.</p>



If this key is not used, **sysidtool** uses broadcast to attempt to bind to either a NIS+ or NIS server; if a name server is not on the local subnet, which is possible for NIS+, the bind will fail, automatic configuration of the name service will fail, and an interactive screen is displayed, prompting the user to specify the name service.

The **ns** entry has the form:

```
ns=[server] : [nameservice] [(netmask)]
```

where:

*server* the name of a server that will provide a name service to bind to  
*nameservice* the name service (**nis**, **nisplus**, or **none**);  
*netmask* a series of four numbers separated by periods that specifies which portion of an IP address is the network part, and which is the host part.

The **ns** keyword can be set in **add\_install\_client** or by Host Manager.

#### EXAMPLES

Here is an example of an entry in the **bootparams** file:

```
client1 root=server1:/export/client1/root \  
swap=server1:/export/client1/swap \  
domain=bldg1.workco.com  
root=server2:/export/client2/root ns=:nis  
root=server2:/export/client2/root ns=watson:  
root=server2:/export/client2/root ns=mach:nisplus(255.255.255.0)
```

#### FILES

/etc/bootparams

#### SEE ALSO

**x86 only**

**rpc.bootparamd**(1M), **sysidtool**(1M), **nsswitch.conf**(4)  
**rpld**(1M)

#### NOTES

Solaris diskless clients use the identifiers "root", "swap", and "dump" to look up the pathnames for the root filesystem, a swap area, and a dump area, respectively. These are the only identifiers meaningful for SPARC diskless booting clients.

For x86 booting clients, the additional keyword identifiers "numbootfiles," "bootfile," and "bootaddr" are used (see **rpld**(1M)).

<b>NAME</b>	cdtoc – CD-ROM table of contents file
<b>DESCRIPTION</b>	<p>The table of contents file, <b>.cdtoc</b>, is an ASCII file that describes the contents of a CD-ROM or other software distribution media. It resides in the top-level directory of the file system on a slice of a CD-ROM. It is independent of file system format, that is, the file system on the slice can be either UFS or HSFS.</p> <p>Each entry in the <b>.cdtoc</b> file is a line that establishes the value of a parameter in the following form:</p> <p style="text-align: center;"><i>PARAM=value</i></p> <p>Blank lines and comments (lines preceded by a pound-sign, “#”) are also allowed in the file. Parameters are grouped by product, with the beginning of a product defined by a line of the form:</p> <p style="text-align: center;"><i>PRODNAME=value</i></p> <p>Each product is expected to consist of one or more software packages that are stored together in a subdirectory on the distribution media. There can be any number of products described within the file. There is no required order in which the parameters must be specified, except that the parameters must be grouped by product and the <i>PRODNAME</i> parameter must appear first in the list of parameters for each product specified. Each parameter is described below. All of the parameters are required for each product.</p> <p><i>PRODNAME</i>      The full name of the product. This must be unique within the <b>.cdtoc</b> file and is preferably unique across all possible products. This value may contain white space. The length of this value is limited to <b>256</b> ASCII characters; other restrictions may apply (see below).</p> <p><i>PRODVERS</i>      The version of the product. The value can contain any combination of letters, numbers, or other characters. This value may contain white space. The length of this value is limited to <b>256</b> ASCII characters; other restrictions may apply (see below).</p> <p><i>PRODDIR</i>        The name of the top-level directory containing the product. This name should be relative to the top-level directory of the distribution media, for example, <b>Solaris_2.6/Product</b>. The number of path components in the name is limited only by the system’s maximum path name length, which is 1024 ASCII characters. Any single component is limited to <b>256</b> ASCII characters. This value cannot contain white space.</p> <p>The lengths of the values of <i>PRODNAME</i> and <i>PRODVERS</i> are further constrained by the fact that the initial install programs and <b>swmtool(1M)</b> concatenate these values to produce the full product name. <b>swmtool(1M)</b> concatenates the two values (inserting a space) to produce the name displayed in its software selection menu, for example, <b>Solaris 2.6</b>. For unbundled products the combined length of the values of <i>PRODNAME</i> and <i>PRODVERS</i> must not exceed <b>256</b> ASCII characters.</p> <p>When you install OS services with Solstice Host Manager, directories for diskless clients and Autoclient systems are created by constructing names derived from a concatenation of the values of <i>PRODNAME</i>, <i>PRODVERS</i>, and client architecture, for example,</p>

**/export/exec/Solaris\_2.x\_sparc.all/usr/platform.** The length of the component containing the product name and version must not exceed **256** ASCII characters. Thus, for products corresponding to bundled OS releases (for example, Solaris 2.4), the values of *PRODNAME* and *PRODDIR* are effectively restricted to lengths much less than **256**.

The initial install programs and **swmtool(1M)** use the value of the *PRODDIR* macro in the **.cdtoc** file to indicate where packages can be found.

#### EXAMPLES

Here is a sample **.cdtoc** file:

```
#
# .cdtoc file -- Online product family CD
#
PRODNAME=Online DiskSuite
PRODVERS=2.0
PRODDIR=Online_DiskSuite_2.0
#
PRODNAME=Online Backup
PRODVERS=2.0
PRODDIR=Online_Backup_2.0
```

This example corresponds to the following directory layout on a CD-ROM partition:

```
/.cdtoc
/Online_DiskSuite_2.0
  ./SUNWmddr.c
  ./SUNWmddr.m
  ./SUNWmddu
/Online_Backup_2.0
  ./SUNWhsm
```

The bundled release of Solaris 2.6 includes the following **.cdtoc** file:

```
PRODNAME=Solaris
PRODVERS=2.6
PRODDIR=Solaris_2.6/Product
```

This file corresponds to the following directory layout on slice 0 of the Solaris 2.6 product CD:

```
/.cdtoc
/Solaris_2.6/Product
  ./SUNWaccr
  ./SUNWaccu
  ./SUNWadmap
  .
  .
  .
  ./SUNWutool
```

**SEE ALSO** | **swmtool(1M), clustertoc(4), packagetoc(4), pkginfo(4)**

<b>NAME</b>	clustertoc – cluster table of contents description file
<b>DESCRIPTION</b>	<p>The cluster table of contents file, <b>.clustertoc</b>, is an ASCII file that describes a hierarchical view of a software product. A <b>.clustertoc</b> file is required for the base OS product. The file resides in the top-level directory containing the product.</p> <p>The hierarchy described by <b>.clustertoc</b> can be of arbitrary depth, although the initial system installation programs assume that it has three levels. The hierarchy is described bottom-up, with the packages described in <b>.packagetoc</b> at the lowest layer. The next layer is the <i>cluster</i> layer which collects packages into functional units. The highest layer is the <i>meta-cluster</i> layer which collects packages and clusters together into typical configurations.</p> <p>The hierarchy exists to facilitate the selection or deselection of software for installation at varying levels of granularity. Interacting at the package level gives the finest level of control over what software is to be installed.</p> <p>Each entry in the <b>.clustertoc</b> file is a line that establishes the value of a parameter in the following form:</p> <p style="text-align: center;"><i>PARAM=value</i></p> <p>A line starting with a pound-sign, “#”, is considered a comment and is ignored.</p> <p>Parameters are grouped by cluster or meta-cluster. The start of a cluster description is defined by a line of the form:</p> <p style="text-align: center;"><i>CLUSTER=value</i></p> <p>The start of a meta-cluster description is defined by a line of the form:</p> <p style="text-align: center;"><i>METACLUSTER=value</i></p> <p>There is no order implied or assumed for specifying the parameters for a (meta-)cluster with the exception of the <i>CLUSTER</i> or <i>METACLUSTER</i> parameter, which must appear first and the <i>END</i> parameter which must appear last.</p> <p>Each parameter is described below. All of the parameters are mandatory.</p> <p><b>CLUSTER</b>      The cluster identifier (for example, SUNWCacc). The identifier specified must be unique within the package and cluster identifier namespace defined by a product’s <b>.packagetoc</b> and <b>.clustertoc</b> files. The identifiers used are subject to the same constraints as those for package identifiers. These constraints are (from <b>pkginfo(4)</b>):</p> <p style="padding-left: 2em;">“All characters in the abbreviation must be alphanumeric and the first may not be numeric. The abbreviation is limited to a maximum length of nine characters. <b>install</b>, <b>new</b>, and <b>all</b> are reserved abbreviations.”</p> <p>A cluster must be described before another cluster or meta-cluster may refer to it.</p>

**METACLUSTER**

The metacluster identifier (for example, *SUNWCprog*). The identifier specified must be unique within the package and cluster identifier namespace defined by a product's **.packagetoc** and **.clustertoc** files. The identifiers used are subject to the same constraints as those for package identifiers. These constraints are (from **pkginfo(4)**):

“All characters in the abbreviation must be alphanumeric and the first may not be numeric. The abbreviation is limited to a maximum length of nine characters. **install**, **new**, and **all** are reserved abbreviations.”

Meta-clusters *cannot* contain references to other meta-clusters.

**NAME**

The full name of the (meta-)cluster. The length of the name string supplied may not exceed 256 characters.

**VENDOR**

The name of the (meta-)cluster's vendor. The length of the vendor string supplied may not exceed 256 characters.

**VERSION**

The version of the (meta-)cluster. The length of the version string supplied may not exceed 256 characters.

**DESC**

An informative textual description of the (meta-)cluster's contents. The length of the description supplied may not exceed 256 characters. The text should contain no newlines.

**SUNW\_CSRMEMBER**

Indicates that the package or cluster is a part of the (meta-) cluster currently being described. The value specified is the identifier of the package or cluster. There may be an arbitrary number of *SUNW\_CSRMEMBER* parameters per (meta-)cluster.

**SUNW\_CSRMBRIFF**

Indicates that the package is to be included dynamically in the (meta-)cluster currently being described. The value of this parameter must follow the following format:

`SUNW_CSRMBRIFF=(<test> <test_arc>)<package>`

This line will be converted into a *SUNW\_CSRMEMBER* entry at media installation time if the test provided matches the platform on which the media is being installed. There may be zero or more *SUNW\_CSRMBRIFF* parameters per (meta-)cluster.

`SUNW_CSRMBRIFF=(<test> <value>)<package>`

where the *<test>* is either the builtin test of "platform" or a shell script which returns shell true (0) or shell false (1) depending on the tests being performed in the script. *<value>* is passed to the test as the first argument and can be used to create a script that tests for multiple hardware objects. Finally *<package>* is the package that will be included in the final **.clustertoc** file as a *SUNW\_CSRMEMBER*. See **parse\_dynamic\_clustertoc(1M)** for more information about the scripts.

**EXAMPLES**

The following is an example of a cluster description in a **.clustertoc** file.

```

CLUSTER=SUNWCacc
NAME=System Accounting
DESC=System accounting utilities
VENDOR=Sun Microsystems, Inc.
VERSION=7.2
SUNW_CSRMEMBER=SUNWaccr
SUNW_CSRMEMBER=SUNWaccu
END

```

The following is an example of a meta-cluster description in a **.clustertoc** file.

```

METACLUSTER=SUNWCreq
NAME=Core System Support
DESC=A pre-defined software configuration consisting of the minimum
required software for a standalone, non-networked workstation.
VENDOR=Sun Microsystems, Inc.
VERSION=2.x
SUNW_CSRMEMBER=SUNWadmr
SUNW_CSRMEMBER=SUNWcar
SUNW_CSRMEMBER=SUNWCcs
SUNW_CSRMEMBER=SUNWCcg6
SUNW_CSRMEMBER=SUNWCdfb
SUNW_CSRMEMBER=SUNWkvm
SUNW_CSRMEMBER=SUNWCnis
SUNW_CSRMEMBER=SUNWowdv
SUNW_CSRMEMBER=SUNWter
END

```

The following is an example of a meta-cluster description with a dynamic cluster entry as indicated by the use of the **SUNW\_CSRMBRIFF** parameter entries.

```

METACLUSTER=SUNWCprog
NAME=Developer System Support
DESC=A pre-defined software configuration consisting of the
typical software used by software developers.
VENDOR=Sun Microsystems, Inc.
VERSION=2.5
SUNW_CSRMEMBER=SUNWCadm
SUNW_CSRMBRIFF=(smcc.dctoc tcx)SUNWCtcx
SUNW_CSRMBRIFF=(smcc.dctoc leo)SUNWCleo
SUNW_CSRMBRIFF=(smcc.dctoc sx)SUNWCsx
...
END

```

**SEE ALSO**

**parse\_dynamic\_clustertoc(1M)**, **cdtoc(4)**, **order(4)**, **packagetoc(4)**, **pkginfo(4)**

**NOTES**

The current implementation of the initial system installation programs depend on the **.clustertoc** describing three required meta-clusters for the base OS product:

- SUNWCall* contains all of the software packages in the OS distribution.
- SUNWUser* contains the typical software packages for an end-user of the OS distribution.
- SUNWCreq* contains the bare-minimum packages required to boot and configure the OS to the point of running a multi-user shell.



<b>NAME</b>	compver – compatible versions file
<b>DESCRIPTION</b>	<p><b>compver</b> is an ASCII file used to specify previous versions of the associated package which are upward compatible. It is created by a package developer.</p> <p>Each line of the file specifies a previous version of the associated package with which the current version is backward compatible.</p> <p>Since some packages may require installation of a specific version of another software package, compatibility information is extremely crucial. Consider, for example, a package called "A" which requires version "1.0" of application "B" as a prerequisite for installation. If the customer installing "A" has a newer version of "B" (version 1.3), the <b>compver</b> file for "B" must indicate that "1.3" is compatible with version "1.0" in order for the customer to install package "A".</p>
<b>EXAMPLES</b>	<p>A sample <b>compver</b> file is shown below:</p> <p><b>Version 1.3</b> <b>Version 1.0</b></p>
<b>SEE ALSO</b>	<p><b>pkginfo(4)</b> <i>Application Packaging Developer's Guide</i></p>
<b>NOTES</b>	<p>The comparison of the version string disregards white space and tabs. It is performed on a word-by-word basis. Thus, "Version 1.3" and "Version 1.3" would be considered the same.</p> <p>The entries in the <b>compver</b> file must match the values assigned to the <b>VERSION</b> parameter in the <b>pkginfo(4)</b> files.</p>

<b>NAME</b>	copyright – copyright information file
<b>DESCRIPTION</b>	<b>copyright</b> is an ASCII file used to provide a copyright notice for a package. The text may be in any format. The full file contents (including comment lines) are displayed on the terminal at the time of package installation.
<b>SEE ALSO</b>	<i>Application Packaging Developer's Guide</i>

<b>NAME</b>	core – core image file
<b>DESCRIPTION</b>	<p>The operating system writes out a core image of a process when it is terminated due to the receipt of some signals. The core image is called <b>core</b> and is written in the process's working directory (provided it can be; normal access controls apply). A process with an effective user ID different from the real user ID will not produce a core image.</p> <p>The core file contains all the process information pertinent to debugging: contents of hardware registers, process status, and process data. The format of a core file is object file specific.</p> <p>For ELF executable programs (see <b>a.out(4)</b>), the core file generated is also an ELF file, containing ELF program and file headers. The <b>e_type</b> field in the file header has type <b>ET_CORE</b>. The program header contains an entry for every segment that was part of the process address space, including shared library segments. The contents of the writable segments are also part of the core image.</p> <p>The program header of an ELF core file also contains entries for two <b>NOTE</b> segments, each containing several note entries as described below. The note entry header and core file note type (<b>n_type</b>) definitions are contained in <b>&lt;sys/elf.h&gt;</b>. The first <b>NOTE</b> segment exists for binary compatibility with old programs that deal with core files. It contains structures defined in <b>&lt;sys/old_procfs.h&gt;</b>. New programs should recognize and skip this <b>NOTE</b> segment, advancing instead to the new <b>NOTE</b> segment. The old <b>NOTE</b> segment will be deleted from core files in a future release.</p> <p>The old <b>NOTE</b> segment contains the following entries. Each has entry name "CORE" and presents the contents of a system structure:</p> <p><b>prpsinfo_t</b>  <b>n_type</b>: <b>NT_PRPSINFO</b>. This entry contains information of interest to the <b>ps(1)</b> command, such as process status, CPU usage, "nice" value, controlling terminal, user-ID, process-ID, the name of the executable, and so forth. The <b>prpsinfo_t</b> structure is defined in <b>&lt;sys/old_procfs.h&gt;</b>.</p> <p><b>char array</b>  <b>n_type</b>: <b>NT_PLATFORM</b>. This entry contains a string describing the specific model of the hardware platform on which this core file was created. This information is the same as provided by <b>sysinfo(2)</b> when invoked with the command <b>SI_PLATFORM</b>.</p> <p><b>auxv_t array</b>  <b>n_type</b>: <b>NT_AUXV</b>. This entry contains the array of <b>auxv_t</b> structures that was passed by the operating system as startup information to the dynamic linker. Auxiliary vector information is defined in <b>&lt;sys/auxv.h&gt;</b>.</p> <p>Following these entries, for each <i>light-weight process</i> (LWP) in the process, the old <b>NOTE</b> segment contains an entry with a <b>prstatus_t</b> structure, plus other optionally-present entries describing the LWP, as follows:</p>

**prstatus\_t**

**n\_type:** NT\_PRSTATUS. This structure contains things of interest to a debugger from the operating system, such as the general registers, signal dispositions, state, reason for stopping, process-ID, and so forth. The **prstatus\_t** structure is defined in `<sys/old_procfs.h>`.

**prfpregset\_t**

**n\_type:** NT\_PRFPREG. This entry is present only if the LWP used the floating-point hardware. It contains the floating-point registers. The **prfpregset\_t** structure is defined in `<sys/old_procfs.h>`.

**gwindows\_t**

**n\_type:** NT\_GWINDOWS. This entry is present only on a SPARC machine and only if the system was unable to flush all of the register windows to the stack. It contains all of the unspilled register windows. The **gwindows\_t** structure is defined in `<sys/regset.h>`.

**prxregset\_t**

**n\_type:** NT\_PRXREG. This entry is present only if the machine has extra register state associated with it. It contains the extra register state. The **prxregset\_t** structure is defined in `<sys/old_procfs.h>`.

The new NOTE segment contains the following entries. Each has entry name "CORE" and presents the contents of a system structure:

**psinfo\_t**

**n\_type:** NT\_PSINFO. This structure contains information of interest to the **ps(1)** command, such as process status, CPU usage, "nice" value, controlling terminal, user-ID, process-ID, the name of the executable, and so forth. The **psinfo\_t** structure is defined in `<sys/procfs.h>`.

**pstatus\_t**

**n\_type:** NT\_PSTATUS. This structure contains things of interest to a debugger from the operating system, such as pending signals, state, process-ID, and so forth. The **pstatus\_t** structure is defined in `<sys/procfs.h>`.

**char array**

**n\_type:** NT\_PLATFORM. This entry contains a string describing the specific model of the hardware platform on which this core file was created. This information is the same as provided by **sysinfo(2)** when invoked with the command **SI\_PLATFORM**.

**auxv\_t array**

**n\_type:** NT\_AUXV. This entry contains the array of **auxv\_t** structures that was passed by the operating system as startup information to the dynamic linker. Auxiliary vector information is defined in `<sys/auxv.h>`.

Following these entries, for each LWP in the process, the new NOTE segment contains an entry with an **lwpsinfo\_t** structure plus an entry with an **lwpstatus\_t** structure, plus other optionally-present entries describing the LWP, as follows:

**lwpsinfo\_t**

**n\_type:** NT\_LWPSINFO. This structure contains information of interest to the **ps(1)** command, such as LWP status, CPU usage, "nice" value, LWP-ID, and so forth. The **lwpsinfo\_t** structure is defined in **<sys/procfs.h>**.

**lwpstatus\_t**

**n\_type:** NT\_LWPSTATUS. This structure contains things of interest to a debugger from the operating system, such as the general registers, the floating point registers, state, reason for stopping, LWP-ID, and so forth. The **lwpstatus\_t** structure is defined in **<sys/procfs.h>**.

**gwindows\_t**

**n\_type:** NT\_GWINDOWS. This entry is present only on a SPARC machine and only if the system was unable to flush all of the register windows to the stack. It contains all of the unspilled register windows. The **gwindows\_t** structure is defined in **<sys/regset.h>**.

**prxregset\_t**

**n\_type:** NT\_PRXREG. This entry is present only if the machine has extra register state associated with it. It contains the extra register state. The **prxregset\_t** structure is defined in **<sys/old\_procfs.h>**.

The size of the core file created by a process may be controlled by the user (see **getrlimit(2)**).

**SEE ALSO**

**adb(1)**, **gcore(1)**, **ps(1)**, **crash(1M)**, **getrlimit(2)**, **setuid(2)**, **sysinfo(2)**, **elf(3E)**, **a.out(4)**, **proc(4)**, **signal(5)**

*ANSI C Programmer's Guide*

<b>NAME</b>	default_fs, fs – specify the default file system type for local or remote file systems
<b>DESCRIPTION</b>	<p>When file system administration commands have both specific and generic components (for example, <b>fsck(1M)</b>), the file system type must be specified. If it is not explicitly specified using the <b>-F <i>FSType</i></b> command line option, the generic command looks in <b>/etc/vfstab</b> in order to determine the file system type, using the supplied raw or block device or mount point. If the file system type can not be determined by searching <b>/etc/vfstab</b>, the command will use the default file system type specified in either <b>/etc/default/fs</b> or <b>/etc/dfs/dfstypes</b>, depending on whether the file system is local or remote.</p> <p>The default local file system type is specified in <b>/etc/default/fs</b> by a line of the form <b>LOCAL=<i>fstype</i></b> (for example, <b>LOCAL=ufs</b>). The default remote file system type is determined by the first entry in the <b>/etc/dfs/dfstypes</b> file.</p> <p>File system administration commands will determine whether the file system is local or remote by examining the specified device name. If the device name starts with “/” (slash), it is considered to be local; otherwise it is remote.</p> <p>The default file system types can be changed by editing the default files with a text editor.</p>
<b>FILES</b>	<p><b>/etc/vfstab</b> list of default parameters for each file system</p> <p><b>/etc/default/fs</b> the default local file system type</p> <p><b>/etc/dfs/dfstypes</b> the default remote file system type</p>
<b>SEE ALSO</b>	<b>fsck(1M)</b> , <b>fstypes(4)</b> , <b>vfstab(4)</b>

<b>NAME</b>	defaultrouter – configuration file for default router(s)	
<b>SYNOPSIS</b>	<b>/etc/defaultrouter</b>	
<b>DESCRIPTION</b>	<p>The <b>/etc/defaultrouter</b> file defines the default routers the system will use. The format of the file is as follows:</p> <p>The <b>/etc/defaultrouter</b> file can contain the hostnames or IP addresses of one or more default routers, separated by white space. If you use hostnames, each hostname must also be listed in the local <b>/etc/hosts</b> file, because no name services are running at the time that this script is run.</p> <p>Lines beginning with the “#” character are treated as comments.</p> <p>The default routes listed in this file replace those added by the kernel during diskless booting. An empty <b>/etc/defaultrouter</b> file will cause the default route added by the kernel to be deleted.</p>	
<b>FILES</b>	<b>/etc/defaultrouter</b>	Configuration file containing the hostnames or IP addresses of one or more default routers.
<b>SEE ALSO</b>	<b>hosts(4)</b>	

<b>NAME</b>	depend – software dependencies file
<b>DESCRIPTION</b>	<p><b>depend</b> is an ASCII file used to specify information concerning software dependencies for a particular package. The file is created by a software developer.</p> <p>Each entry in the <b>depend</b> file describes a single software package. The instance of the package is described after the entry line by giving the package architecture and/or version. The format of each entry and subsequent instance definition is:</p> <pre style="margin-left: 40px;">type pkg name       (arch)version       (arch)version       ...</pre> <p>The fields are:</p> <p><i>type</i> Defines the dependency type. Must be one of the following characters:</p> <ul style="list-style-type: none"> <li><b>P</b> Indicates a prerequisite for installation; for example, the referenced package or versions must be installed.</li> <li><b>I</b> Implies that the existence of the indicated package or version is incompatible.</li> <li><b>R</b> Indicates a reverse dependency. Instead of defining the package's own dependencies, this designates that another package depends on this one. This type should be used only when an old package does not have a <b>depend</b> file, but relies on the newer package nonetheless. Therefore, the present package should not be removed if the designated old package is still on the system since, if it is removed, the old package will no longer work.</li> </ul> <p><i>pkg</i> Indicates the package abbreviation.</p> <p><i>name</i> Specifies the full package name.</p> <p><i>(arch)version</i> Specifies a particular instance of the software. A version name cannot begin with a left parenthesis. The instance specifications, both <i>(arch)</i> and <i>version</i>, are completely optional, but each <i>(arch)version</i> pair must begin on a new line that begins with white space. A null version set equates to any version of the indicated package.</p>
<b>EXAMPLES</b>	<p>Here is a sample <b>depend</b> file:</p> <pre style="margin-left: 40px;">#ident "@(#)pkg.compat:depend 1.1" <b>P nsu</b>      <b>Networking Support Utilities</b> <b>P inet</b>     <b>Internet Utilities</b> <b>P sys</b>      <b>System Header Files</b> <b>P src_compat</b> <b>Source Compatibility Files</b></pre>
<b>SEE ALSO</b>	<i>Application Packaging Developer's Guide</i>



<b>NAME</b>	device_allocate – device_allocate file												
<b>SYNOPSIS</b>	<b>/etc/security/device_allocate</b>												
<b>DESCRIPTION</b>	<p>The <b>device_allocate</b> file contains mandatory access control information about each physical device. Each device is represented by a one line entry of the form:</p> <p style="text-align: center;"><i>device-name;device-type;reserved;reserved;alloc;device-exec</i></p> <p>where</p> <table border="0" style="margin-left: 2em;"> <tr> <td style="vertical-align: top;"><i>device-name</i></td> <td>This is an arbitrary ASCII string naming the physical device. This field contains no embedded white space or non-printable characters.</td> </tr> <tr> <td style="vertical-align: top;"><i>device-type</i></td> <td>This is an arbitrary ASCII string naming the generic device type. This field identifies and groups together devices of like type. This field contains no embedded white space or non-printable characters.</td> </tr> <tr> <td style="vertical-align: top;">reserved</td> <td>This field is reserved for future use.</td> </tr> <tr> <td style="vertical-align: top;">reserved</td> <td>This field is reserved for future use.</td> </tr> <tr> <td style="vertical-align: top;"><i>alloc</i></td> <td>This field contains an arbitrary string which controls whether or not a device is allocatable. If the field contains only an asterisk (*), the device is <i>not</i> allocatable. Otherwise, the device may be allocated and deallocated in the normal fashion.</td> </tr> <tr> <td style="vertical-align: top;"><i>device-exec</i></td> <td>This is the physical device's data purge program to be run any time the device is acted on by <b>allocate(1M)</b>. This is to ensure that all usable data is purged from the physical device before it is reused. This field contains the filename of a program in <b>/etc/security/lib</b> or the full pathname of a cleanup script provided by the system administrator.</td> </tr> </table> <p>The <b>device_allocate</b> file is an ASCII file that resides in the <b>/etc/security</b> directory.</p> <p>Lines in <b>device_allocate</b> can end with a <code>`\`</code> to continue an entry on the next line.</p> <p>Comments may also be included. A <code>#</code> makes a comment of all further text until the next NEWLINE not immediately preceded by a <code>`\`</code>.</p> <p>Leading and trailing blanks are allowed in any of the fields.</p> <p>The <b>device_allocate</b> file must be created by the system administrator before device allocation is enabled.</p> <p>The <b>device_allocate</b> file is owned by root, with a group of sys, and a mode of 0644.</p>	<i>device-name</i>	This is an arbitrary ASCII string naming the physical device. This field contains no embedded white space or non-printable characters.	<i>device-type</i>	This is an arbitrary ASCII string naming the generic device type. This field identifies and groups together devices of like type. This field contains no embedded white space or non-printable characters.	reserved	This field is reserved for future use.	reserved	This field is reserved for future use.	<i>alloc</i>	This field contains an arbitrary string which controls whether or not a device is allocatable. If the field contains only an asterisk (*), the device is <i>not</i> allocatable. Otherwise, the device may be allocated and deallocated in the normal fashion.	<i>device-exec</i>	This is the physical device's data purge program to be run any time the device is acted on by <b>allocate(1M)</b> . This is to ensure that all usable data is purged from the physical device before it is reused. This field contains the filename of a program in <b>/etc/security/lib</b> or the full pathname of a cleanup script provided by the system administrator.
<i>device-name</i>	This is an arbitrary ASCII string naming the physical device. This field contains no embedded white space or non-printable characters.												
<i>device-type</i>	This is an arbitrary ASCII string naming the generic device type. This field identifies and groups together devices of like type. This field contains no embedded white space or non-printable characters.												
reserved	This field is reserved for future use.												
reserved	This field is reserved for future use.												
<i>alloc</i>	This field contains an arbitrary string which controls whether or not a device is allocatable. If the field contains only an asterisk (*), the device is <i>not</i> allocatable. Otherwise, the device may be allocated and deallocated in the normal fashion.												
<i>device-exec</i>	This is the physical device's data purge program to be run any time the device is acted on by <b>allocate(1M)</b> . This is to ensure that all usable data is purged from the physical device before it is reused. This field contains the filename of a program in <b>/etc/security/lib</b> or the full pathname of a cleanup script provided by the system administrator.												

**EXAMPLES**

Declare that physical device st0 is a type **st**. **st** is allocatable, and the script used to clean the device after running **deallocate(1M)** is named **/etc/security/lib/st\_clean**.

```

# scsi tape
st0;\
  st;\
  reserved;\
  reserved;\
  alloc;\
  /etc/security/lib/st_clean;\

```

Declare that physical device fd0 is of type **fd**. **fd** is allocatable, and the script used to clean the device after running **deallocate(1M)** is named **/etc/security/lib/fd\_clean**.

```

# floppy drive
fd0;\
  fd;\
  reserved;\
  reserved;\
  alloc;\
  /etc/security/lib/fd_clean;\

```

Note that making a device allocatable means that you need to allocate and deallocate them to use them (with **allocate(1M)** and **deallocate(1M)**). If a device is allocatable, there will be an asterisk (\*) in the *alloc* field, and one can use the device without allocating and deallocating it.

**FILES** **/etc/security/device\_allocate** Contains list of allocatable devices

**SEE ALSO** **allocate(1M)**, **bsmconv(1M)**, **deallocate(1M)**, **list\_devices(1M)**

**NOTES** 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.

<b>NAME</b>	device.cfinfo – devconfig configuration files
<b>SYNOPSIS</b>	<b>device.cfinfo</b>
<b>DESCRIPTION</b>	<p><b>device.cfinfo</b> files pass information about device configuration to the <b>devconfig(1M)</b> program. They allow <b>devconfig(1M)</b> to provide the user with valid ranges for device attributes.</p> <p><b>devconfig(1M)</b> associates a device with its <b>cfinfo</b> file by name. For example, the device <b>logi</b> for the Logitech Bus Mouse has the <b>devconfig(1M)</b> configuration file <b>logi.cfinfo</b> associated with it in the <b>DEVCONFIGHOME</b> directory. <b>DEVCONFIGHOME</b> is <b>/usr/lib/devconfig</b> by default and may be set in the user's environment.</p> <p>Below is a yaccish grammar of a <b>cfinfo</b> file:</p> <pre> <b>cfinfo_file:</b>      <b>cfinfo_devspec EOF</b>                     ;  <b>cfinfo_devspec:</b>  <b>cfinfo_spec_list SEMICOLON</b>                     ;  <b>cfinfo_spec_list:</b> <b>cfinfo_spec  </b>                     <b>cfinfo_spec_list cfinfo_spec</b>                     ;  <b>cfinfo_spec:</b>     <b>comment  </b>                     <b>attr_value_pair NEWLINE</b>                     ;  <b>comment:</b>        <b>POUNDSIGN  </b>                     <b>POUNDSIGN STRING</b>                     ;  <b>attr_value_pair:</b> <b>ATTR_NAME EQUALS STRING  </b>                     <b>ATTR_OWNAME EQUALS STRING</b>                     <b>ATTR_TITLE EQUALS STRING  </b>                     <b>ATTR_CATEGORY EQUALS STRING  </b>                     <b>ATTR_INSTANCE EQUALS STRING  </b>                     <b>ATTR_CLASS EQUALS STRING  </b>                     <b>ATTR_TYPE EQUALS STRING  </b>                     <b>ATTR_REAL EQUALS STRING  </b>                     <b>ATTR_AUTO EQUALS STRING  </b>                     <b>NAME EQUALS value_spec_string</b>                     ;  <b>value_spec_string:</b> <b>QUOTE value_spec QUOTE</b>                     ; </pre>

```

value_spec:      value_type COMMA value_list
                   ;

value_type:      | /* EMPTY */
                   TYPE_NUMERIC |
                   TYPE_STRING |
                   TYPE_VAR
                   ;

value_list:      integer_value_list |
                   string_value_list
                   ;

integer_value_list: INTEGER |
                   INTEGER COLON INTEGER |
                   INTEGER COMMA integer_value_list
                   ;

string_value_list: STRING |
                   STRING COMMA string_value_list
                   ;

ATTR_NAME      name          # device name specified in driver.conf
ATTR_CLASS    class         # device class specified in driver.conf
ATTR_TYPE     type          # device type specified in OWconfig
ATTR_OWNAME   __owname__     # device name specified in OWconfig
ATTR_TITLE    __title__      # device title displayed by devconfig
ATTR_CATEGORY __category__  # device category
ATTR_INSTANCE __instance__  # device unit
ATTR_REAL     __real__       # attributes to write to driver.conf
ATTR_AUTO     __auto__       # self-identifying device attribute
TYPE_NUMERIC  numeric       # precedes an integer value list
TYPE_STRING   string        # precedes a string values list
TYPE_VAR     var           # precedes a variable specification

```

The first value in a **value\_list** is the default value picked by **devconfig(1M)** for the attribute. An attribute name of the form **\_\_name\_\_** is used internally by **devconfig(1M)**. Number ranges are specified as **n1:n2**. An internal attribute of the type **var** specifies a configurable portion of a real attribute. (See examples below.) Certain internal attributes have an expanded form when displayed. These attributes are listed in the file **abbreviations** in **DEVCONFIGHOME**. The file **abbreviations** also includes a list of name mappings for certain category names. If the **\_\_real\_\_** attribute is present, only the attribute names it specifies are written to a driver.conf file. Otherwise, all non-internal attributes are written.

## EXAMPLES

Here is the device configuration file **logi.cfinfo** for the LOGITECH bus mouse. The driver configuration file for this device is called **logi.conf**.

```

name="logi"
    __owname__="pointer:0"
    __title__="Logitech bus mouse"

    __category__="pointer"

    class="sysbus"
    type="LOGI-B"
    buttons="var,__nbuttons__"
    __nbuttons__="numeric,2:3"
    dev="/dev/logi"

    intr="numeric,1","var,__irq__"
    __irq__="numeric,2:5"

    __real__="name","class","intr"
    ;

```

The driver name for the LOGITECH Bus Mouse is **logi**. The device name in **OWconfig** (see the *OpenWindows Desktop Reference Manual*) is **pointer:0**. The device category is **pointer**; the device category is displayed as **pointing devices**, however, since there is a category mapping for **pointer** in the **abbreviations** file. The device class is **sysbus** as specified in the file `/kernel/drv/classes`. A device of class **owin** does not have a device driver associated with it. The device IPL is **1**. The device IRQ is substituted by the variable **\_\_irq\_\_** and has a range of **2** to **5**. A name mapping for **\_\_irq\_\_** exists in **abbreviations** and so **\_\_irq\_\_** is displayed as **Interrupt (IRQ)**. The device attributes written to **logi.conf** are **name**, **class**, and **intr** as specified by the **\_\_real\_\_** entry.

The resulting entry in **logi.conf** is:

```
name="logi" class="sysbus" intr=1,2;
```

The resulting entry in **OWconfig** is:

```
type="LOGI-B" buttons=3 dev="/dev/logi" class="owin" name="pointer:0";
```

Here is an example of a self-identifying device.

```

name="lp"
    __title__="Parallel printer port"
    __category__="lp"

    class="sysbus"

    __auto__="string,true"
    ;

```

The driver for the parallel port automatically identifies it, and **devconfig(1M)** treats this device as self-identifying.

**FILES****abbreviations****ATTRIBUTES**

See **attributes(5)** for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Architecture	x86

**SEE ALSO**

**devconfig(1M)**, **driver.conf(4)**, **attributes(5)**

*OpenWindows Desktop Reference Manual*

<b>NAME</b>	device_maps – device_maps file						
<b>SYNOPSIS</b>	<code>/etc/security/device_maps</code>						
<b>DESCRIPTION</b>	<p>The <b>device_maps</b> file contains access control information about each physical device. Each device is represented by a one line entry of the form:</p> <pre style="margin-left: 40px;"><i>device-name</i> : <i>device-type</i> : <i>device-list</i> :</pre> <p>where</p> <table border="0" style="margin-left: 40px;"> <tr> <td style="vertical-align: top;"><i>device-name</i></td> <td>This is an arbitrary ASCII string naming the physical device. This field contains no embedded white space or non-printable characters.</td> </tr> <tr> <td style="vertical-align: top;"><i>device-type</i></td> <td>This is an arbitrary ASCII string naming the generic device type. This field identifies and groups together devices of like type. This field contains no embedded white space or non-printable characters.</td> </tr> <tr> <td style="vertical-align: top;"><i>device-list</i></td> <td>This is a list of the device special files associated with the physical device. This field contains valid device special file path names separated by white space.</td> </tr> </table> <p>The <b>device_maps</b> file is an ASCII file that resides in the <code>/etc/security</code> directory. Lines in <b>device_maps</b> can end with a <code>'\'</code> to continue an entry on the next line. Comments may also be included. A <code>'#'</code> makes a comment of all further text until the next NEWLINE not immediately preceded by a <code>'\'</code>. Leading and trailing blanks are allowed in any of the fields. The <b>device_maps</b> file must be created by the system administrator before device allocation is enabled. This file is owned by root, with a group of <code>sys</code>, and a mode of <code>0644</code>.</p>	<i>device-name</i>	This is an arbitrary ASCII string naming the physical device. This field contains no embedded white space or non-printable characters.	<i>device-type</i>	This is an arbitrary ASCII string naming the generic device type. This field identifies and groups together devices of like type. This field contains no embedded white space or non-printable characters.	<i>device-list</i>	This is a list of the device special files associated with the physical device. This field contains valid device special file path names separated by white space.
<i>device-name</i>	This is an arbitrary ASCII string naming the physical device. This field contains no embedded white space or non-printable characters.						
<i>device-type</i>	This is an arbitrary ASCII string naming the generic device type. This field identifies and groups together devices of like type. This field contains no embedded white space or non-printable characters.						
<i>device-list</i>	This is a list of the device special files associated with the physical device. This field contains valid device special file path names separated by white space.						
<b>EXAMPLES</b>	<pre style="margin-left: 40px;"># scsi tape st1:\ rmt:\ /dev/rst21 /dev/nrst21 /dev/rst5 /dev/nrst5 /dev/rst13 \ /dev/nrst13 /dev/rst29 /dev/nrst29 /dev/rmt/1l /dev/rmt/1m \ /dev/rmt/1 /dev/rmt/1h /dev/rmt/1u /dev/rmt/1ln /dev/rmt/1mn \ /dev/rmt/1n /dev/rmt/1hn /dev/rmt/1un /dev/rmt/1b /dev/rmt/1bn:\</pre>						
<b>FILES</b>	<code>/etc/security/device_maps</code>						
<b>SEE ALSO</b>	<code>allocate(1M)</code> , <code>bsmconv(1M)</code> , <code>deallocate(1M)</code> , <code>dminfo(1M)</code> , <code>list_devices(1M)</code>						
<b>NOTES</b>	The functionality described in this man page is available only if the Basic Security Module (BSM) has been enabled. See <code>bsmconv(1M)</code> for more information.						

<b>NAME</b>	dfstab – file containing commands for sharing resources across a network
<b>DESCRIPTION</b>	<p><b>dfstab</b> resides in directory <code>/etc/dfs</code> and contains commands for sharing resources across a network. <b>dfstab</b> gives a system administrator a uniform method of controlling the automatic sharing of local resources.</p> <p>Each line of the <b>dfstab</b> file consists of a <b>share(1M)</b> command. The <b>dfstab</b> file can be read by the shell to share all resources. System administrators can also prepare their own shell scripts to execute particular lines from <b>dfstab</b>.</p> <p>The contents of <b>dfstab</b> are executed automatically when the system enters run-level 3.</p>
<b>SEE ALSO</b>	<b>share(1M)</b> , <b>shareall(1M)</b>



**NAME** dhcp – file containing default parameter values for the location and type of the databases used by the DHCP service

**DESCRIPTION** The **dhcp** file resides in directory **/etc/default** and contains parameters for specifying the type and location of DHCP service databases.

The **dhcp** file format is ASCII; comment lines begin with the crosshatch (#) character. Parameters consist of a keyword followed by an equals (=) sign followed by the parameter value, of the form:

*Keyword=Value*

Two parameters are currently supported:

<i>Keyword</i>	<i>Value</i>
<b>RESOURCE</b>	Can be either <b>nisplus</b> or <b>files</b>
<b>PATH</b>	Path to data files

The value of the **PATH** keyword is specified as an absolute path for the **files** resource, or a fully-qualified directory for the **nisplus** resource. The preferred method of modifying the **dhcp** file is through use of the **dhcpconfig(1M)** command.

**SEE ALSO** **dhcpconfig(1M)**, **in.dhcpd(1M)**

**NAME** dhcp\_network – dhcp network DHCP database

**DESCRIPTION** The **dhcp network** database is used to map a Dynamic Host Configuration Protocol (DHCP) client's client identifier to an IP address and the associated configuration parameters of that address. This database is located by the DHCP server at runtime upon receipt of a BOOTP request.

The **dhcp network** databases can exist as NIS+ tables or ASCII files. Since the format of the file could change, the preferred method of managing the **dhcp network** databases is through the use of the **pentadm(1M)** command.

Each entry in a **dhcp network** database has the form:

Client_ID	Flags	Client_IP	Server_IP	Lease	Macro	#Comment
-----------	-------	-----------	-----------	-------	-------	----------

The fields are defined as follows:

**Client\_ID** The client identifier field, **Client\_ID**, is an ASCII hexadecimal representation of the unique octet string which identifies the DHCP client. The number of characters in this field must be an even number, with a maximum length of 64 characters. Valid characters are **0 - 9** and **A-F**. Entries with values of **00** are freely available for dynamic allocation to requesting clients.

BOOTP clients are identified by the concatenation of the network's hardware type (as defined by RFC 1340, titled "Assigned Numbers") and the client's hardware address. For example, the following BOOTP client has a hardware type of **'01'** (10mb ethernet) and a hardware address of **8:0:20:11:12:b7**, so its client identifier would be:

**010800201112B7**

**Flags** The **Flags** field is a numeric bit field which can have a combination of the following values:

- 0 (DYNAMIC)** Evaluation of the **Lease** field is turned on.
- 1 (PERMANENT)** Evaluation of the **Lease** field is turned off (lease is permanent).
- 2 (MANUAL)** This entry has a manual client ID binding (cannot be reclaimed by DHCP server). Client will not be allocated another address.
- 4 (UNUSABLE)** When set, this value means that either through ICMP echo or client DECLINE, this address has been found to be unusable. Can also be used by the network administrator to *prevent* a certain client from booting, if used in conjunction with the **MANUAL** flag.
- 8 (BOOTP)** This entry is reserved for allocation to BOOTP clients only.

<b>Client_IP</b>	The <b>Client_IP</b> field holds the IP address for this entry. This value must be unique in the database.
<b>Server_IP</b>	This field holds the IP address of the DHCP server which <i>owns</i> this client IP address, and thus is responsible for initial allocation to a requesting client.
<b>Lease</b>	This numeric field holds the entry's absolute lease expiration time, and is in seconds since <b>January 1, 1970</b> . It can be decimal, or hexadecimal (if <b>0x</b> prefixes number). The special value <b>-1</b> is used to denote a permanent lease.
<b>Macro</b>	This ASCII text field contains the <b>dhcptab</b> macro name used to look up this entry's configuration parameters in the <b>dhcptab(4)</b> database.
<b>Comment</b>	This ASCII text field contains an optional comment.

#### TREATISE ON LEASES

This section describes how the DHCP/BOOTP server calculates a client's configuration lease using information contained in the **dhcptab(4)** and **dhcp network** databases. The server consults the **LeaseTim** and **LeaseNeg** symbols in the **dhcptab**, and the **Flags** and **Lease** fields of the chosen **dhcp network** database record.

The server first examines the **Flags** field for the identified **dhcp network** record. If the **PERMANENT** flag is on, then the client's lease is considered permanent.

If the **PERMANENT** flag is not on, then the server checks if the client's lease as represented by the **Lease** field in the **dhcp network** record has expired. If not, then the server checks if the client has requested a new lease. If the **LeaseNeg** symbol has not been included in the client's **dhcptab** parameters, then the client's requested lease extension is ignored, and the lease is set to be the time remaining as shown by the **Lease** field. If the **LeaseNeg** symbol *has* been included, then the server will extend the client's lease to the value it requested if this requested lease is less than or equal to the current time plus the value of the client's **LeaseTim dhcptab** parameter.

If the client's requested lease is greater than policy allows (value of **LeaseTim**), then the client is given a lease equal to the current time plus the value of **LeaseTim**. If **LeaseTim** is not set, then the default **LeaseTim** value is one hour.

For more information about the **dhcptab** symbols discussed in this section, see **dhcptab(4)**.

#### EXAMPLES

- 1) The following **dhcp network** database entry is free for dynamic allocation. The IP address for this entry is **10.0.0.5**, the IP address of the DHCP server that can initially allocate this address is **10.0.0.1**, the lease expires **754012553**, or **Mon Nov 22 18:55:53 1993**, and the **dhcctab** macro associated with this entry is called **10netnis**:

```
00  0  10.0.0.5 10.0.0.1 754012553  10netnis
```

- 2) The following entry shows a manually administered entry for client ID **01000C0EFA4A**, which has a permanent lease (that is, **MANUAL | PERMANENT == 3**):

```
01000C0EFA4A 3 10.0.0.25 10.0.0.1 -1 10netnis
```

- 3) The following entry shows a **MANUAL** entry which has been marked as **UNUSABLE** (that is, **MANUAL | UNUSABLE == 6**):

```
0408072097C9F 6 10.0.0.26 10.0.0.1 764258362 10netdns
```

- 4) The following entry for IP address **10.0.0.27** shows a previously unused, **DYNAMIC** entry which uses **dhcptab** macro **10netnis** and is owned by DHCP server **10.0.0.2**:

```
00 0 10.0.0.27 10.0.0.2 0 10netnis
```

- 5) The following entry is reserved for BOOTP clients:

```
00 08 10.0.0.27 10.0.0.3 0 10netnis
```

**FILES**

**/var/dhcp/NNN\_NNN\_NNN\_NNN**

Where **NNN\_NNN\_NNN\_NNN** are database file(s) or NIS+ tables(s).

**/var/dhcp/dhcptab**

file or NIS+ table

**SEE ALSO**

**dhcpconfig(1M)**, **dhtadm(1M)**, **in.dhcpd(1M)**, **pntadm(1M)**, **dhcptab(4)**

Reynolds, J. and J. Postel, *Assigned Numbers*, STD 2, RFC 1340, USC/Information Sciences Institute, July 1992,

**NAME** dhcptab – DHCP configuration parameter table

**DESCRIPTION** The **dhcptab** macro table allows network administrators to organize groups of configuration parameters as macro definitions, which can then be further used in the definition of other useful macros. These macros can be configured such that the DHCP server will return their values to DHCP and BOOTP clients.

The preferred method of managing the **dhcptab** macro table is through the use of the **dhtadm**(1M) utility. The syntax described in the balance of this manual page is intended for informational purposes.

**Syntax of the dhcptab Table**

The syntax of the **dhcptab** table is as follows:

Comments begin with the cross-hatch (#) character in the first position on the line and end with a carriage return. Lines can be continued by escaping the carriage return character with a backslash (\) character.

**dhcptab** records contain three (3) fields:

Name	Type	Value
------	------	-------

The fields are defined as follows:

**Name** This field identifies the record and is used as the search key into the **dhcptab** table. A **Name** must consist of ASCII characters. If the record is of type **Macro**, then the length is limited to 64 characters. If the record is of type **Symbol**, then the length is limited to 8 characters.

**Type** This field specifies the type of record. Currently, there are only two legal values for **Type**:

**m (Macro)** This record is a DHCP macro definition.

**s (Symbol)** This record is a DHCP symbol definition. It is used to define vendor and site-specific options.

**Value** This field contains the value for the specified type of record. For the **macro** type, the value will consist of a series of symbol=value pairs, separated by the colon (:) character. For the **symbol** type, the value will consist of a series of fields, separated by a comma (,), which define a symbol's characteristics. Once defined, a symbol can be used in macro definitions.

**Symbol Characteristics**

The fields describing the characteristics of a symbol are as follows:

Context	Code	Type	Granularity	Maximum
---------	------	------	-------------	---------

These fields are defined as follows:

**Context** This field defines the context in which the symbol definition is to be used. It can have three values:

**Extend** This symbol defines a standard option, codes from 77-127. The use of this symbol type is for adding new standard options added since the release of the **dhcp** server.

<b>Site</b>	This symbol defines a site-specific option, codes 128-254.										
<b>Vendor=Client Class ...</b>	This symbol defines a vendor-specific option, codes 1-254. The Vendor context takes ASCII string arguments which identify the client class that this vendor option is associated with. Multiple client class names can be specified, separated by white space. Only those clients whose client class matches one of these values will see this option.										
<b>Code</b>	This field specifies the option code number associated with this symbol. Valid values are 128-254 for site-specific options, and 1-254 for vendor-specific options.										
<b>Type</b>	This field defines the type of data expected as a value for this symbol. Legal values are: <table> <tr> <td><b>ASCII</b></td> <td>NVT ASCII text. Value is enclosed in double-quotes (").</td> </tr> <tr> <td><b>BOOLEAN</b></td> <td>No value is associated with this data type. Presence of symbols of this type denote boolean <b>TRUE</b>, whereas absence denotes <b>FALSE</b>.</td> </tr> <tr> <td><b>IP</b></td> <td>Dotted decimal form of an Internet address.</td> </tr> <tr> <td><b>NUMBER</b></td> <td>An unsigned number with a supported granularity of <b>1</b>, <b>2</b>, <b>4</b>, and <b>8</b> octets.</td> </tr> <tr> <td><b>OCTET</b></td> <td>Uninterpreted ASCII representation of binary data. The client identifier is one example of an octet string.</td> </tr> </table>	<b>ASCII</b>	NVT ASCII text. Value is enclosed in double-quotes (").	<b>BOOLEAN</b>	No value is associated with this data type. Presence of symbols of this type denote boolean <b>TRUE</b> , whereas absence denotes <b>FALSE</b> .	<b>IP</b>	Dotted decimal form of an Internet address.	<b>NUMBER</b>	An unsigned number with a supported granularity of <b>1</b> , <b>2</b> , <b>4</b> , and <b>8</b> octets.	<b>OCTET</b>	Uninterpreted ASCII representation of binary data. The client identifier is one example of an octet string.
<b>ASCII</b>	NVT ASCII text. Value is enclosed in double-quotes (").										
<b>BOOLEAN</b>	No value is associated with this data type. Presence of symbols of this type denote boolean <b>TRUE</b> , whereas absence denotes <b>FALSE</b> .										
<b>IP</b>	Dotted decimal form of an Internet address.										
<b>NUMBER</b>	An unsigned number with a supported granularity of <b>1</b> , <b>2</b> , <b>4</b> , and <b>8</b> octets.										
<b>OCTET</b>	Uninterpreted ASCII representation of binary data. The client identifier is one example of an octet string.										
<b>Granularity</b>	This value specifies how many objects of <b>Type</b> define a single <b>instance</b> of the symbol value. For example, the static route option is defined to be a variable list of routes. Each route consists of two IP addresses, so the <b>Type</b> is defined to be <b>IP</b> , and the data's granularity is defined to be <b>2</b> IP addresses.										
<b>Maximum</b>	This value specifies the maximum items of <b>Granularity</b> which are permissible in a definition using this symbol. For example, there can only be one IP address specified for a subnet mask, so the <b>Maximum</b> number of items in this case is one ( <b>1</b> ). A <b>Maximum</b> value of zero ( <b>0</b> ) means that a variable number of items is permitted.										

The following example defines a site-specific option called **MystatRt**, of code **130**, type **IP**, and granularity **2**, and a **Maximum** of **0**. This definition corresponds to the internal definition of the static route option (**StaticRt**).

**MystatRt s Site,130,IP,2,0**

**Macro Definitions**

The following example illustrates a macro defined using the **MystatRt** site option symbol just defined:

**10netnis m :MystatRt=3.0.0.0 10.0.0.30:**

Macro records can be specified in the **Macro** field in **dhcp network** databases (see **dhcp\_network(4)**), which will bind particular macro definitions to specific IP addresses.

If present, four macro definitions are consulted by the DHCP server to determine the options that are returned to the requesting client:

Client Class	Network	IP Address	Client Identifier
--------------	---------	------------	-------------------

These macros are processed as follows:

- Client Class** A macro called by the ASCII representation of the client class is searched for in the **dhcptab**. If found, then its symbol/value pairs will be selected for delivery to the client. This mechanism permits the network administrator to select configuration parameters to be returned to all clients of the same class.
- Network** A macro named by the dotted Internet form of the network address of the client's network (for example, **10.0.0.0**) is searched for in the **dhcptab**. If found, then its symbol/value pairs will be combined with those of the **Client Class** macro. If a symbol exists in both macros, then the **Network** macro value overrides the value defined in the **Client Class** macro. This mechanism permits the network administrator to select configuration parameters to be returned to all clients on the same network.
- IP Address** This macro is specified in the **dhcp network** database for the record assigned to the requesting client. If this macro is found in the **dhcptab**, then its symbol/value pairs will be combined with those of the **Client Class** macro and the **Network** macro. This mechanism permits the network administrator to select configuration parameters to be returned to clients using a particular IP address. It can also be used to deliver a macro defined to include "server-specific" information by including this macro definition in all **dhcp network** database entries owned by a specific server.
- Client Identifier** A macro called by the ASCII representation of the client's client identifier is searched for in the **dhcptab**. If found, its symbol/value pairs are combined to the sum of the **Client Class**, **Network**, and **IP Address** macros. Any symbol collisions are replaced with those specified in the client identifier macro. This mechanism permits the network administrator to select configuration parameters to be returned to a particular client, regardless of what network that client is connected to.

**Internal Symbol  
Names**

The following table maps the available internal symbol names to RFC-2132 options:

<i>Symbol</i>	<i>Code</i>	<i>Description</i>
<b>Subnet</b>	<b>1</b>	Subnet Mask, dotted Internet address (IP).
<b>UTCoffst</b>	<b>2</b>	Coordinated Universal time offset (seconds).
<b>Router</b>	<b>3</b>	List of Routers, IP.
<b>Timeserv</b>	<b>4</b>	List of RFC-868 servers, IP.
<b>IEN116ns</b>	<b>5</b>	List of IEN 116 name servers, IP.
<b>DNSserv</b>	<b>6</b>	List of DNS name servers, IP.
<b>Logserv</b>	<b>7</b>	List of MIT-LCS UDP log servers, IP.
<b>Cookie</b>	<b>8</b>	List of RFC-865 cookie servers, IP.
<b>Lprserv</b>	<b>9</b>	List of RFC-1179 line printer servers, IP.
<b>Impress</b>	<b>10</b>	List of Imagen Impress servers, IP.
<b>Resource</b>	<b>11</b>	List of RFC-887 resource location servers, IP.
<b>Hostname</b>	<b>12</b>	Client's hostname, value from hosts database.
<b>Bootsize</b>	<b>13</b>	Number of 512 octet blocks in boot image, NUMBER.
<b>Dumpfile</b>	<b>14</b>	Path where core image should be dumped, ASCII.
<b>DNSdmain</b>	<b>15</b>	DNS domain name, ASCII.
<b>Swapserv</b>	<b>16</b>	Client's swap server, IP.
<b>Rootpath</b>	<b>17</b>	Client's Root path, ASCII.
<b>ExtendP</b>	<b>18</b>	Extensions path, ASCII.
<b>IpFwdF</b>	<b>19</b>	IP Forwarding Enable/Disable, NUMBER.
<b>NLrouteF</b>	<b>20</b>	Non-local Source Routing, NUMBER.
<b>PFilter</b>	<b>21</b>	Policy Filter, IP,IP.
<b>MaxIpSiz</b>	<b>22</b>	Maximum datagram Reassembly Size, NUMBER.
<b>IpTTL</b>	<b>23</b>	Default IP Time to Live, (1=<x<=255), NUMBER.
<b>PathTO</b>	<b>24</b>	RFC-1191 Path MTU Aging Timeout, NUMBER.
<b>PathTbl</b>	<b>25</b>	RFC-1191 Path MTU Plateau Table, NUMBER.
<b>MTU</b>	<b>26</b>	Interface MTU, x>=68, NUMBER.
<b>SameMtuF</b>	<b>27</b>	All Subnets are Local, NUMBER.
<b>Broadcst</b>	<b>28</b>	Broadcast Address, IP.
<b>MaskDscF</b>	<b>29</b>	Perform Mask Discovery, NUMBER.
<b>MaskSupF</b>	<b>30</b>	Mask Supplier, NUMBER.
<b>RDiscvyF</b>	<b>31</b>	Perform Router Discovery, NUMBER.
<b>RSolictS</b>	<b>32</b>	Router Solicitation Address, IP.
<b>StaticRt</b>	<b>33</b>	Static Route, Double IP (network router).
<b>TrailerF</b>	<b>34</b>	Trailer Encapsulation, NUMBER.
<b>ArpTimeO</b>	<b>35</b>	ARP Cache Time out, NUMBER.
<b>EthEncap</b>	<b>36</b>	Ethernet Encapsulation, NUMBER.
<b>TcpTTL</b>	<b>37</b>	TCP Default Time to Live, NUMBER.
<b>TcpKaInt</b>	<b>38</b>	TCP Keepalive Interval, NUMBER.
<b>TcpKaGbF</b>	<b>39</b>	TCP Keepalive Garbage, NUMBER.
<b>NISdmain</b>	<b>40</b>	NIS Domain name, ASCII.
<b>NISservs</b>	<b>41</b>	List of NIS servers, IP.
<b>NTPservs</b>	<b>42</b>	List of NTP servers, IP.
<b>NetBNms</b>	<b>44</b>	List of NetBIOS Name servers, IP.



<b>NetBDsts</b>	<b>45</b>	List of NetBIOS Distribution servers, IP.
<b>NetBNdT</b>	<b>46</b>	NetBIOS Node type (1=B-node, 2=P, 4=M, 8=H)
<b>NetBScop</b>	<b>47</b>	NetBIOS scope, ASCII.
<b>XFontSrv</b>	<b>48</b>	List of X Window Font servers, IP.
<b>XDispMgr</b>	<b>49</b>	List of X Window Display managers, IP.
<b>LeaseTim</b>	<b>51</b>	Lease Time Policy, (-1 = PERM), NUMBER.
<b>Message</b>	<b>56</b>	Message to be displayed on client, ASCII.
<b>T1Time</b>	<b>58</b>	Renewal (T1) time, NUMBER.
<b>T2Time</b>	<b>59</b>	Rebinding (T2) time, NUMBER.
<b>NW_dmain</b>	<b>62</b>	NetWare/IP Domain Name, ASCII.
<b>NWIPOpts</b>	<b>63</b>	NetWare/IP Options, OCTET (unknown type).
<b>NIS+dom</b>	<b>64</b>	NIS+ Domain name, ASCII.
<b>NIS+serv</b>	<b>65</b>	NIS+ servers, IP.
<b>TFTPSrvN</b>	<b>66</b>	TFTP server hostname, ASCII.
<b>OptBootF</b>	<b>67</b>	Optional Bootfile path, ASCII.
<b>MblIPAgnt</b>	<b>68</b>	Mobile IP Home Agent, IP.
<b>SMTPserv</b>	<b>69</b>	Simple Mail Transport Protocol Server, IP.
<b>POP3serv</b>	<b>70</b>	Post Office Protocol (POP3) Server, IP.
<b>NNTPserv</b>	<b>71</b>	Network News Transport Proto. (NNTP) Server, IP.
<b>WWWservs</b>	<b>72</b>	Default WorldWideWeb Server, IP.
<b>Fingersv</b>	<b>73</b>	Default Finger Server, IP.
<b>IRCservs</b>	<b>74</b>	Internet Relay Chat Server, IP.
<b>STservs</b>	<b>75</b>	StreetTalk Server, IP.
<b>STDAservs</b>	<b>76</b>	StreetTalk Directory Assist. Server, IP.
<b>BootFile</b>	N/A	File to Boot, ASCII.
<b>BootSrvA</b>	N/A	Boot Server, IP.
<b>BootSrvN</b>	N/A	Boot Server Hostname, ASCII.
<b>LeaseNeg</b>	N/A	Lease is Negotiable Flag, (Present=TRUE)
<b>Include</b>	N/A	Include listed macro values in this macro.

**EXAMPLES**

Below is an example **dhcptab** file, illustrating the concepts described above:

```
#
# PCNFS vendor options. First define them, then use them in
# our Client Class macro definition to establish proper context.
#

# SolarNet framework servers. Note that this symbol is valid for two
# client classes, "SUNW.PCNFS.5.1" and "SUNW.PCNFSPRO.1.1".
SNadmfw s Vendor=SUNW.PCNFS.5.1.1 SUNW.PCNFSPRO.1.1,1,ASCII,1,0

# PCNFS servers. Note that two client classes are specified for
# this symbol.
Pcnfsd s Vendor=SUNW.PCNFS.5.1.1 SUNW.PCNFSPRO.1.1,2,IP,1,0

# NFS Read and Write sizes. Unsigned shorts.
```

```

SNnfsRd s Vendor=SUNW.PCNFS.5.1.1 SUNW.PCNFSPRO.1.1,4,NUMBER,2,1
SNnfsWr s Vendor=SUNW.PCNFS.5.1.1 SUNW.PCNFSPRO.1.1,5,NUMBER,2,1

# NFS Timeout in 1/10's of a second. An unsigned short.
SNnfsTim s Vendor=SUNW.PCNFS.5.1.1 SUNW.PCNFSPRO.1.1,6,NUMBER,2,1

# NFS Retries, an unsigned short.
SNnfsTry s Vendor=SUNW.PCNFS.5.1.1 SUNW.PCNFSPRO.1.1,7,NUMBER,2,1

# PC-Admin login script file.
SNClgin s Vendor=SUNW.PCNFS.5.1.1 SUNW.PCNFSPRO.1.1,8,ASCII,1,0

# PC-Admin logout script file.
SNClgout s Vendor=SUNW.PCNFS.5.1.1 SUNW.PCNFSPRO.1.1,9,ASCII,1,0

# PC-Admin script server.
SNCserv s Vendor=SUNW.PCNFS.5.1.1 SUNW.PCNFSPRO.1.1,10,IP,1,0

# Path to PC-Admin scripts on server.
SNCpath s Vendor=SUNW.PCNFS.5.1.1 SUNW.PCNFSPRO.1.1,11,ASCII,1,0

# PC-Admin Boot script file.
SNCboot s Vendor=SUNW.PCNFS.5.1.1 SUNW.PCNFSPRO.1.1,12,ASCII,1,0

# Timezone (TZ)
SN_TZ s Vendor=SUNW.PCNFS.5.1.1 SUNW.PCNFSPRO.1.1,13,ASCII,1,0

# Site specific option.
SiteTest s Site,128,IP,1,1

# PCNFS client class. This option will automatically be returned
# to clients specifying "SUNW.PCNFS.5.1.1" as their Client Class.
# Predefined, Site, or vendor symbols can be used in this definition.
# However, note that vendor symbols used here whose Client Class does not
# match will be omitted in the response to the client.
SUNW.PCNFS.5.1.1 m \
    :SNadmfw="doppelbock pilsner": \
    :Pcnfsd=10.0.5.26 10.0.5.5 10.0.4.1: \
    :SNnfsRd=1024:SNnfsWr=8192: \
    :SNnfsTim=56:SNnfsTry=6: \
    :Impress=10.0.0.254:

# Set the locale. EST's offset is 18000 seconds. Note also the use
# of the SN_TZ (which will overwrite UTCoffst for SUNW.PCNFS.5.1.1 and
# SUNW.PCNFSPRO.1.1 clients).

```

```

Locale m \
    :UTCoffst=18000:SN_TZ="EST5EDT":

# Netbios node type is broadcast (1).
NetBIOS m \
    :NetBNms=10.0.5.1 10.0.4.1:NetBNdT=0x1: \
    :NetBDsts=10.0.5.5 10.0.5.6 10.0.4.2: \
    :NetBScop="NB.This.Is.A.Nis.DOMAIN":

# This macro includes the definitions for Locale and NetBIOS.
# Lease is renegotiable, and the maximum lease a client can request
# is 2 hours (7200 seconds)
#
# Note that this macro definition includes the SUNW.PCNFS.5.1.1 and
# SUNW.PCNFSPRO.1.1 Vendor symbol for SolarNet login script file name.
# Only those clients whose Client Class is SUNW.PCNFS.5.1.1 will see
# this value.
5netnis m \
    :Subnet=255.255.255.0:Router=10.0.5.26 10.0.5.27: \
    :Include=Locale:SNCpath="/opt/SUNWpcnet/1.5/site/pcnfs": \
    :SNCboot="boot.snc":SNCserv=10.0.5.26:Timeserv=10.0.5.5: \
    :NISdmain="This.Is.A.Nis.DOMAIN":NISservs=10.0.5.210: \
    :Message="NIS client, Welcome to the 5 net.": \
    :SiteTest=1.0.0.0:LeaseTim=7200:LeaseNeg:Include=NetBIOS: \
    :SNClogin="login.snc":

# This macro defines a short lease - only 5 minutes! Note the use
# of the pcnfsd vendor option here. Note also that the server will
# return the client's hostname by consulting the hosts database for
# the value.
15netnis m \
    :Subnet=255.255.255.0:Router=10.0.15.226: \
    :Include=Locale:SNCpath="/opt/solarnet":SNCboot="site.snc": \
    :SNCserv=10.0.15.226:Timeserv=10.0.5.5: \
    :NISdmain="Another.Nis.Domain.COM":NISservs=10.0.15.6: \
    :Message="NIS client, Welcome to the 15 net.": \
    :LeaseTim=300:LeaseNeg:Pcnfsd=10.0.15.226:Hostname:

5netdns m \
    :Subnet=255.255.255.0:Router=10.0.5.26 10.0.5.26: \
    :SNCserv=10.0.5.26:SNCpath="/opt/SUNWpcnet/site/pcnfs": \
    :SNCboot="boot.snc":Include=Locale:Timeserv=10.0.5.5: \
    :DNSdmain="East.Sun.COM":DNSserv=10.0.15.6 15.0.1.15: \
    :Message="DNS client, Welcome to the 5 net.":LeaseNeg:

```

```

# This macro is named by a client's client identifier. Its options
# will be combined with those of the Client Class macro
# and per network macro, if defined. Regardless of where this client
# boots, these options will follow it!
010800C0EE0E4C m \
    :Impress=10.0.20.55:

```

**FILES** /var/dhcp/dhcptab file or NIS+ table.

**SEE ALSO** dhcpconfig(1M), dhtadm(1M), in.dhcpd(1M), dhcp\_network(4)

Alexander, S., and R. Droms, *DHCP Options and BOOTP Vendor Extensions*, RFC 2132, Silicon Graphics, Inc., Bucknell University, March 1997.

Droms, R., *Interoperation Between DHCP and BOOTP*, RFC 1534, Bucknell University, October 1993.

Droms, R., *Dynamic Host Configuration Protocol*, RFC 2131, Bucknell University, March 1997.

Wimer, W., *Clarifications and Extensions for the Bootstrap Protocol*, RFC 1542, Carnegie Mellon University, October 1993.

<b>NAME</b>	dhcptags – DHCP option mnemonic mapping table
<b>DESCRIPTION</b>	For the most part, parameters (henceforth referred to as options) returned to the client by the DHCP/BOOTP protocol are encoded in the so-called <i>vendor</i> field of the <b>BOOTP</b> packet. Each option is identified numerically, and also carries a length specifier. The purpose of <b>dhcptags</b> is to indentify the type of each option, to label each with a short mnemonic text string for use by <b>dhcpinfo</b> (1), and to give a longer textual description.
<b>OPTIONS</b>	
<b>General Options</b>	Options defined by DHCP are of three general types: <ul style="list-style-type: none"> <li><i>Standard</i>            All client and server DHCP implementations agree on the semantics. These are administered by the Internet Naming Authority (IANA). These options are numbered from 1 to 127.</li> <li><i>Site-specific</i>        Within a specific site, all client and server implementations agree as to the semantics. However, at another site the type and meaning of the option may be quite different. These options are numbered from 128 to 254.</li> <li><i>Vendor-specific</i>      Each vendor may define 256 options unique to that vendor. The vendor is identified within a DHCP packet by the "Vendor Class" option (#60). An option with a specific numeric identifier belonging to one vendor will, in general, have a type and semantics different from that of a different vendor. Vendor options are "super-encapsulated" into the <b>vendor</b> field (#43); within a specific DHCP packet there may be several instances of option #43.</li> </ul>
<b>Pseudo Options</b>	As well as the three general types, the Solaris DHCP implementation defines certain "pseudo" options, numbered from 512 upward. These are a convenient method for referring to items which either correspond to fixed fields in the <b>BOOTP</b> packet (such as the <b>siaddr</b> field) or which, though not options themselves, are used in constructing valid options (for example, the <i>home directory</i> used in constructing the exact path to a boot image).  In general, the agent (see <b>dhcpagent</b> (1M)) knows little if anything about the semantics of any of the first three kinds of option, except for the subnet mask and broadcast address. Its only duty is to acquire and store this data and to make it available to other interested parties (see <b>dhcpinfo</b> (1)). The responsibility for understanding and using the data rests with these third parties. Pseudo tags, on the contrary, have a specific meaning to <b>dhcpagent</b> (1M), and consequently it is meaningless to add to this list. The only useful edit that can be performed on the pseudo tags is to change the textual description or the mnemonic.
<b>USAGE</b>	Blank lines and those whose first non-whitespace character is '#' are ignored. Data entries are written one per line and have five fields. An individual entry cannot be continued onto another line.

The fields are (in order):

- Tag number
- Mnemonic identifier
- Vendor class
- Data type

(One from the following case insensitive values):

**byte**  
**octet**  
**int1**        A 1-byte value  
**int2**        A 2-byte value  
**int4**        A 4-byte value  
**string**      A printable character string  
**ip**            An IP address  
**iplist**      A list of IP addresses  
**int2list**    A list of 2-byte values  
**opaque**     An array of 1-byte values  
**boolean**    Either **true** or **false**

- Long name

#### Standard Option List

Table of Standard Tags			
<i>Tag Number</i>	<i>Identifier</i>	<i>Data Type</i>	<i>Description</i>
1	NetMask	ip	Subnet mask
2	UTCoffst	time	Time offset from GMT
3	Router	iplist	IP addresses of routers
4	Timesrv	iplist	IP addresses of time servers
5	IEN116ns	iplist	IP addresses of IEN=116 name servers
6	DNSserv	iplist	IP addresses of domain name servers
7	Logserv	iplist	IP addresses of remote logging servers
8	Cookie	iplist	IP address list of fortune cookie servers
9	Lprserv	iplist	IP address list of print servers
10	Impress	iplist	IP address list of impress servers
11	Resource	iplist	IP address list of RLP servers
12	Hostname	string	hostname (or nodename) of client
13	Bootsize	int16	size (in 512 blocks) of client boot file
14	Dumpfile	string	path name of Merit dump file
15	DNSdmain	string	DNS domain name
16	Swapserv	ip	ip address of swap file server
17	Rootpath	ip	
18	ExtendP	string	
19	IPFwdF	boolean	Enable IP forwarding
20	NLrouteF	boolean	
21	PFilter	iplist	IP address list of policy filter servers
22	MaxIpSiz	int16	Maximum reassembly size of IP datagram
23	IpTTL	byte	IP time-to-live field
24	PathTO	time	PMTU timeout
25	PathTbl	int16list	PMTU plateaus
26	MTU	int16	Maximum transmission unit

<b>Table of Standard Tags</b>			
<i>Tag Number</i>	<i>Identifier</i>	<i>Data Type</i>	<i>Description</i>
27	SameMtuF	boolean	Subnets are local
28	Broadcst	ip	IP broadcast address of interface
29	MaskDscf	boolean	When true perform mask discovery
30	MaskSupF	boolean	When true supply subnet masks
31	RDiscvyF	boolean	Perform route discovery
32	RsolicitS	ip	IP address for router solicitation
33	StaticRt	iplist	Pairs of IP addresses for all static routes
34	TrailerF	boolean	Perform trailer encapsulation
35	ArpTimeO	time	Timeout interval for entry in ARP cache
36	EthEncap	boolean	Perform Ethernet encapsulation
37	TcpTTL	byte	TCP time-to-live
38	TcpKaInt	time	TCP keep alive interval
39	TcpKaGbF	boolean	Send TCP keep alive garbage octet
40	NISdmain	string	NIS domain name
41	NISservs	iplist	IP address list
42	NTPservs	iplist	IP address list of NTP servers
44	NetBNms	iplist	IP address list of NetBios name servers
45	NetBDsts	iplist	IP address list of NetBios DG servers
46	NetBNdT	byte	NetBios node type
47	NetBScop	string	NetBios scope
48	XFontSrv	iplist	IP address list of X font servers
49	XDispMgr	iplist	IP address list of X display managers
50	RequestIP	ip	IP address requested by client
51	LeaseTim	time	Lease duration (secs)
52	Overload	byte	File and/or sname fields overloaded
53	MsgType	byte	DHCP message type
54	ServerIp	ip	IP address of DHCP server selected by client
55	rv	opaque	DHCP options requested by client
56	Message	string	Message from DHCP server to client
57	MaxMsgSz	byte	Maximum BOOTP message size acceptable
58	T1Time	time	DHCP renewal interval
59	T2Time	time	DHCP rebind interval
60	Vendor	string	Client's vendor class
61	ClientID	opaque	Client identifier
62	NW_domain	string	Netware domain
63	NWIPOpts	string	Netware options
64	NIS+dom	string	NIS+ domain name
65	NIS+serv	iplist	IP address list of NIS+ servers
66	TFTPsrvtN	string	Boot file server name
67	OptBootF	string	Path to boot file on boot file server
68	MblIPAgnt	iplist	IP address list of mobile IP home agents
69	SMTPserv	iplist	IP address list of SMTP servers
70	POP3serv	iplist	IP address list of POP servers
71	NNTPserv	iplist	IP address list of NNTP servers
72	WWWsertvs	iplist	IP address list of WWW servers
73	Fingervsv	iplist	IP address list of Finger servers
74	IRCServsv	iplist	IP address list of IRC servers

Table of Standard Tags			
Tag Number	Identifier	Data Type	Description
75	STservs	iplist	IP address list of StreetTalk servers
76	STDAservs	iplist	IP address list of STDA servers
77	UserClass	string	Client's user class

**FILES** /etc/dhcp/dhcptags

**ATTRIBUTES** See **attributes(5)** for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsr

**SEE ALSO** **dhcpinfo(1)**, **dhcpage(1M)**, **attributes(5)**

Alexander, S., and R. Droms, *DHCP Options and BOOTP Vendor Extensions*, RFC 2132, Silicon Graphics, Inc., Bucknell University, March 1997.

Droms, R., *Dynamic Host Configuration Protocol*, RFC 2131, Bucknell University, March 1997.



<b>NAME</b>	dialups – list of terminal devices requiring a dial-up password	
<b>SYNOPSIS</b>	<i>/etc/dialups</i>	
<b>DESCRIPTION</b>	<p><b>dialups</b> is an ASCII file which contains a list of terminal devices that require a dial-up password. A dial-up password is an additional password required of users who access the computer through a modem or dial-up port. The correct password must be entered before the user is granted access to the computer. The set of ports that require a dial-up password are listed in the <b>dialups</b> file.</p> <p>Each entry in the <b>dialups</b> file is a single line of the form:</p> <p style="padding-left: 40px;"><i>terminal-device</i></p> <p>where</p> <p style="padding-left: 40px;"><i>terminal-device</i>    The full path name of the terminal device that will require a dial-up password for users accessing the computer through a modem or dial-up port.</p> <p>The <b>dialups</b> file should be owned by the <b>root</b> user and the <b>root</b> group. The file should have read and write permissions for the owner (<b>root</b>) only.</p>	
<b>EXAMPLES</b>	Here is a sample <b>dialups</b> file:	
	<pre style="padding-left: 40px;">/dev/term/a /dev/term/b /dev/term/c</pre>	
<b>FILES</b>	<i>/etc/d_passwd</i>	dial-up password file
	<i>/etc/dialups</i>	list of dial-up ports requiring dial-up passwords
<b>SEE ALSO</b>	<b>d_passwd</b> (4)	

<b>NAME</b>	dirent – file system independent directory entry
<b>SYNOPSIS</b>	<b>#include &lt;dirent.h&gt;</b>
<b>DESCRIPTION</b>	<p>Different file system types may have different directory entries. The <b>dirent</b> structure defines a file system independent directory entry, which contains information common to directory entries in different file system types. A set of these structures is returned by the <b>getdents(2)</b> system call.</p> <p>The <b>dirent</b> structure is defined:</p> <pre> struct dirent {     ino_t          d_ino;     off_t          d_off;     unsigned short d_reclen;     char           d_name[1]; }; </pre> <p>The <b>d_ino</b> is a number which is unique for each file in the file system. The field <b>d_off</b> is the byte offset of the next, non-empty directory entry in the <i>actual file system directory</i>. The field <b>d_name</b> is the beginning of the character array giving the name of the directory entry. This name is null terminated and may have at most <b>MAXNAMLEN</b> characters. This results in file system independent directory entries being variable length entities. The value of <b>d_reclen</b> is the record length of this entry. This length is defined to be the number of bytes between the current entry and the next one, so that the next structure will be suitably aligned.</p>
<b>SEE ALSO</b>	<b>getdents(2)</b>

<b>NAME</b>	dir_ufs, dir – format of ufs directories				
<b>SYNOPSIS</b>	<pre>#include &lt;sys/param.h&gt; #include &lt;sys/types.h&gt; #include &lt;sys/fs/ufs_fsdir.h&gt;</pre>				
<b>DESCRIPTION</b>	<p>A directory consists of some number of blocks of <b>DIRBLKSIZ</b> bytes, where <b>DIRBLKSIZ</b> is chosen such that it can be transferred to disk in a single atomic operation (for example, 512 bytes on most machines).</p> <p>Each <b>DIRBLKSIZ</b>-byte block contains some number of directory entry structures, which are of variable length. Each directory entry has a <b>struct direct</b> at the front of it, containing its inode number, the length of the entry, and the length of the name contained in the entry. These entries are followed by the name padded to a 4 byte boundary with null bytes. All names are guaranteed null-terminated. The maximum length of a name in a directory is <b>MAXNAMLEN</b>.</p> <pre>#define      DIRBLKSIZ          DEV_BSIZE #define      MAXNAMLEN          256 struct direct {     u_long   d_ino;              /* inode number of entry */     u_short  d_reclen;          /* length of this record */     u_short  d_namlen;          /* length of string in d_name */     char     d_name[MAXNAMLEN + 1]; /* name must be no longer than this */ };</pre>				
<b>ATTRIBUTES</b>	See <b>attributes(5)</b> for a description of the following attributes:				
	<table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>Stability Level</td> <td>Unstable</td> </tr> </tbody> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	Stability Level	Unstable
ATTRIBUTE TYPE	ATTRIBUTE VALUE				
Stability Level	Unstable				
<b>SEE ALSO</b>	<b>fs_ufs(4)</b> , <b>attributes(5)</b>				

<b>NAME</b>	d_passwd – dial-up password file				
<b>SYNOPSIS</b>	/etc/d_passwd				
<b>DESCRIPTION</b>	<p>A dial-up password is an additional password required of users who access the computer through a modem or dial-up port. The correct password must be entered before the user is granted access to the computer.</p> <p><b>d_passwd</b> is an ASCII file which contains a list of executable programs (typically shells) that require a dial-up password and the associated encrypted passwords. When a user attempts to log in on any of the ports listed in the <b>dialups</b> file (see <b>dialups(4)</b>), the login program looks at the user's login entry stored in the <b>passwd</b> file (see <b>passwd(4)</b>), and compares the login shell field to the entries in <b>d_passwd</b>. These entries determine whether the user will be required to supply a dial-up password.</p> <p>Each entry in <b>d_passwd</b> is a single line of the form:</p> <p style="padding-left: 40px;"><i>login-shell:password:</i></p> <p>where</p> <table border="0" style="margin-left: 40px;"> <tr> <td style="padding-right: 20px;"><i>login-shell</i></td> <td>The name of the login program that will require an additional dial-up password.</td> </tr> <tr> <td><i>password</i></td> <td>A 13-character encrypted password. Users accessing the computer through a dial-up port or modem using <i>login-shell</i> will be required to enter this password before gaining access to the computer.</td> </tr> </table> <p><b>d_passwd</b> should be owned by the <b>root</b> user and the <b>root</b> group. The file should have read and write permissions for the owner (<b>root</b>) only.</p> <p>If the user's login program in the <b>passwd</b> file is not found in <b>d_passwd</b> or if the login shell field in <b>passwd</b> is empty, the user must supply the default password. The default password is the entry for <b>/usr/bin/sh</b>. If <b>d_passwd</b> has no entry for <b>/usr/bin/sh</b>, then those users whose login shell field in <b>passwd</b> is empty or does not match any entry in <b>d_passwd</b> will not be prompted for a dial-up password.</p> <p>Dial-up logins are disabled if <b>d_passwd</b> has only the following entry:</p> <p style="padding-left: 40px;"><b>/usr/bin/sh:*</b></p>	<i>login-shell</i>	The name of the login program that will require an additional dial-up password.	<i>password</i>	A 13-character encrypted password. Users accessing the computer through a dial-up port or modem using <i>login-shell</i> will be required to enter this password before gaining access to the computer.
<i>login-shell</i>	The name of the login program that will require an additional dial-up password.				
<i>password</i>	A 13-character encrypted password. Users accessing the computer through a dial-up port or modem using <i>login-shell</i> will be required to enter this password before gaining access to the computer.				
<b>EXAMPLES</b>	<p>Here is a sample <b>d_passwd</b> file:</p> <pre style="padding-left: 40px;">/usr/lib/uucp/uucico:q.mJzTnu8icF0: /usr/bin/csh:6k/7KCFRPNVXg: /usr/bin/ksh:9df/FDf.4jkRt: /usr/bin/sh:41FuGVzGcDJlw:</pre>				

### Generating An Encrypted Password

The **passwd** (see **passwd(1)**) utility can be used to generate the encrypted password for each login program. **passwd** generates encrypted passwords for users and places the password in the **shadow** (see **shadow(4)**) file. Passwords for the **d\_passwd** file will need to be generated by first adding a temporary user id using **useradd** (see **useradd(1M)**), and then using **passwd(1)** to generate the desired password in the **shadow** file. Once the encrypted version of the password has been created, it can be copied to the **d\_passwd** file.

For example:

1. Type **useradd tempuser** and press Return. This creates a user named **tempuser**.
2. Type **passwd tempuser** and press Return. This creates an encrypted password for **tempuser** and places it in the **shadow** file.
3. Find the entry for **tempuser** in the **shadow** file and copy the encrypted password to the desired entry in the **d\_passwd** file.
4. Type **userdel tempuser** and press Return to delete **tempuser**.

These steps must be executed as the **root** user.

<b>FILES</b>	<b>/etc/d_passwd</b>	dial-up password file
	<b>/etc/dialups</b>	list of dial-up ports requiring dial-up passwords
	<b>/etc/passwd</b>	password file
	<b>/etc/shadow</b>	shadow password file

**SEE ALSO** **passwd(1)**, **useradd(1M)**, **dialups(4)**, **passwd(4)**, **shadow(4)**

**WARNINGS** When creating a new dial-up password, be sure to remain logged in on at least one terminal while testing the new password. This ensures that there is an available terminal from which you can correct any mistakes that were made when the new password was added.

<b>NAME</b>	driver.conf – driver configuration files
<b>SYNOPSIS</b>	<b>driver.conf</b>
<b>DESCRIPTION</b>	<p>Driver configuration files pass information about device drivers and their configuration to the system. Most device drivers do not have to have configuration files. Drivers for devices that are self-identifying, such as the SBus devices on many systems, can usually obtain all the information they need from the FCode PROM on the SBus card using the DDI property interfaces. See <b>ddi_prop_get_int(9F)</b> and <b>ddi_prop_lookup(9F)</b> for details.</p> <p>The system associates a driver with its configuration file by name. For example, a driver in <b>/usr/kernel/drv</b> called <b>wombat</b> has the driver configuration file <b>wombat.conf</b> associated with it. By convention, the driver configuration file lives in the same directory as the driver.</p> <p>The syntax of a single entry in a driver configuration file takes one of three forms:</p> <p style="padding-left: 40px;"><b>name="node name" parent="parent name" [property-name=value ...];</b></p> <p>In this form, the parent name can be either a simple nexus driver name to match all instances of that parent/node, or the parent name can be a specific full pathname, beginning with a slash (/) character, identifying a specific instance of a parent bus.</p> <p>Alternatively, the parent can be specified by the type of interface it presents to its children.</p> <p style="padding-left: 40px;"><b>name="node name" class="class name" [property-name=value ...];</b></p> <p>For example, the driver for the SCSI host adapter may have different names on different platforms, but the target drivers can use class <b>scsi</b> to insulate themselves from these differences.</p> <p>Entries of either form above correspond to a device information (devinfo) node in the kernel device tree. Each node has a <i>name</i> which is usually the name of the driver, and a <i>parent</i> name which is the name of the parent devinfo node it will be connected to. Any number of name-value pairs may be specified to create properties on the prototype devinfo node. These properties can be retrieved using the DDI property interfaces (for example, <b>ddi_prop_get_int(9F)</b> and <b>ddi_ddi_prop_lookup(9F)</b>). The prototype devinfo node specification must be terminated with a semicolon (;).</p> <p>The third form of an entry is simply a list of properties.</p> <p style="padding-left: 40px;"><b>[property-name=value ...];</b></p> <p>A property created in this way is treated as global to the driver. It can be overridden by a property with the same name on a particular devinfo node, either by creating one explicitly on the prototype node in the driver.conf file or by the driver.</p> <p>Items are separated by any number of newlines, SPACE or TAB characters.</p> <p>The configuration file may contain several entries to specify different device configurations and parent nodes. The system may call the driver for each possible prototype devinfo node, and it is generally the responsibility of the drivers <b>probe(9E)</b> routine to determine if the hardware described by the prototype devinfo node is really present.</p>

Property names should obey the same naming convention as Open Boot PROM properties, in particular they should not contain at-sign (@), or slash (/) characters. Property values can be decimal integers or strings delimited by double quotes ("). Hexadecimal integers can be constructed by prefixing the digits with **0x**.

A comma separated list of integers can be used to construct properties whose value is an integer array. The value of such properties can be retrieved inside the driver using **ddi\_prop\_lookup\_int\_array(9F)**.

Comments are specified by placing a # character at the beginning of the comment string, the comment string extends for the rest of the line.

## EXAMPLES

Here is a configuration file called **ACME,simple.conf** for a VMEbus frame buffer called **ACME,simple**.

```
#
# Copyright (c) 1993, by ACME Fictitious Devices, Inc.
#
#ident "@(#)ACME,simple.conf 1.3 93/09/09"
```

```
name="ACME,simple" class="vme"
reg=0x7d,0x400000,0x110600;
```

This example creates a prototype devinfo node called **ACME,simple** under all parent nodes of class **vme**. It specifies a property called **reg** that consists of an array of three integers. The **reg** property is interpreted by the parent node, see **vme(4)** for further details.

Here is a configuration file called **ACME,example.conf** for a pseudo device driver called **ACME,example**.

```
#
# Copyright (c) 1993, ACME Fictitious Devices, Inc.
#
#ident "@(#)ACME,example.conf 1.2 93/09/09"
name="ACME,example" parent="pseudo" instance=0
debug-level=1;
```

```
name="ACME,example" parent="pseudo" instance=1;
```

```
whizzy-mode="on";
debug-level=3;
```

This example creates two devinfo nodes called **ACME,example** which will attach below the **pseudo** node in the kernel device tree. The **instance** property is only interpreted by the **pseudo** node, see **pseudo(4)** for further details. A property called **debug-level** will be created on the first devinfo node which will have the value 1. The **example** driver will be able to fetch the value of this property using **ddi\_prop\_get\_int(9F)**.

Two global driver properties are created, **whizzy-mode** (which will have the string value "on") and **debug-level** (which will have the value 3). If the driver looks up the property **whizzy-mode** on either node, it will retrieve the value of the global **whizzy-mode** property ("on"). If the driver looks up the **debug-level** property on the first node, it will retrieve the value of the **debug-level** property on that node (1). Looking up the same property on the second node will retrieve the value of the global **debug-level** property (3).

**SEE ALSO**

**pci(4)**, **pseudo(4)**, **sbus(4)**, **scsi(4)**, **vme(4)**, **probe(9E)**, **ddi\_getlongprop(9F)**, **ddi\_getprop(9F)**, **ddi\_getproplen(9F)**, **ddi\_prop\_op(9F)**

*Writing Device Drivers*

**WARNINGS**

To avoid namespace collisions between multiple driver vendors, it is strongly recommended that the *name* property of the driver should begin with a vendor-unique string. A reasonably compact and unique choice is the vendor over-the-counter stock symbol.



<b>NAME</b>	environ, pref, variables – user-preference variables files for AT&T FACE
<b>SYNOPSIS</b>	<b>\$HOME/pref/.environ</b> <b>\$HOME/pref/.variables</b> <b>\$HOME/FILECABINET/.pref</b> <b>\$HOME/WASTEBASKET/.pref</b>
<b>DESCRIPTION</b>	<p>The <b>.environ</b>, <b>.pref</b>, and <b>.variables</b> files contain variables that indicate user preferences for a variety of operations. The <b>.environ</b> and <b>.variables</b> files are located under the user's <b>\$HOME/pref</b> directory. The <b>.pref</b> files are found under <b>\$HOME/FILECABINET</b>, <b>\$HOME/WASTEBASKET</b>, and any directory where preferences were set via the <b>organize</b> command. Names and descriptions for each variable are presented below. Variables are listed one per line and are of the form <i>variable=value</i>.</p>
<b>.environ Variables</b>	<p>Variables found in <b>.environ</b> include:</p> <p><b>LOGINWIN[1-4]</b>      Windows that are opened when FACE is initialized</p> <p><b>SORTMODE</b>            Sort mode for file folder listings. Values include the following hexadecimal digits:</p> <p style="margin-left: 2em;"><b>1</b>            sorted alphabetically by name</p> <p style="margin-left: 2em;"><b>2</b>            files most recently modified first</p> <p style="margin-left: 2em;"><b>800</b>        sorted alphabetically by object type</p> <p>The values above may be listed in reverse order by ORing the following value:</p> <p style="margin-left: 2em;"><b>1000</b>    list objects in reverse order. For example, a value of <b>1002</b> will produce a folder listing with files LEAST recently modified displayed first. A value of <b>1001</b> would produce a "reverse" alphabetical by name listing of the folder</p> <p><b>DISPLAYMODE</b>        Display mode for file folders. Values include the following hexadecimal digits:</p> <p style="margin-left: 2em;"><b>0</b>            file names only</p> <p style="margin-left: 2em;"><b>4</b>            file names and brief description</p> <p style="margin-left: 2em;"><b>8</b>            file names, description, plus additional information</p> <p><b>WASTEPROMPT</b>        Prompt before emptying wastebasket (yes/no)?</p> <p><b>WASTEDAYS</b>            Number of days before emptying wastebasket</p> <p><b>PRINCMD[1-3]</b>        Print command defined to print files.</p> <p><b>UMASK</b>                Holds default permissions that files will be created with.</p>

<b>.pref Variables</b>	Variables found in <b>.pref</b> are the following: <b>SORTMODE</b> which has the same values as the <b>SORTMODE</b> variable described in <b>.environ</b> above. <b>DISPMODE</b> which has the same values as the <b>DISPLAYMODE</b> variable described in <b>.environ</b> above.
<b>.variable Variables</b>	Variables found in <b>.variables</b> include: <b>EDITOR</b> Default editor <b>PS1</b> shell prompt

<b>NAME</b>	ethers – Ethernet address to hostname database or domain
<b>DESCRIPTION</b>	<p>The <b>ethers</b> file is a local source of information about the (48 bit) Ethernet addresses of hosts on the Internet. The <b>ethers</b> file can be used in conjunction with or instead of other ethers sources, including the NIS maps <b>ethers.byname</b> and <b>ethers.byaddr</b> and the NIS+ table <b>ethers</b>. Programs use the <b>ethers(3N)</b> routines to access this information.</p> <p>The <b>ethers</b> file has one line for each host on an Ethernet. The line has the following format:</p> <p style="text-align: center;"><i>Ethernet-address official-host-name</i></p> <p>Items are separated by any number of SPACE and/or TAB characters. A '#' indicates the beginning of a comment extending to the end of line.</p> <p>The standard form for Ethernet addresses is "x:x:x:x:x" where x is a hexadecimal number between 0 and ff, representing one byte. The address bytes are always in network order. Host names may contain any printable character other than SPACE, TAB, NEWLINE, or comment character.</p>
<b>FILES</b>	<b>/etc/ethers</b>
<b>SEE ALSO</b>	<b>ethers(3N)</b> , <b>hosts(4)</b> , <b>nsswitch.conf(4)</b>

<b>NAME</b>	fd – file descriptor files
<b>DESCRIPTION</b>	<p>These files, conventionally called <code>/dev/fd/0</code>, <code>/dev/fd/1</code>, <code>/dev/fd/2</code>, and so on, refer to files accessible through file descriptors. If file descriptor <code>n</code> is open, these two system calls have the same effect:</p> <pre>fd = open("/dev/fd/n",mode); fd = dup(n);</pre> <p>On these files <code>creat(2)</code> is equivalent to <code>open</code>, and <code>mode</code> is ignored. As with <code>dup</code>, subsequent reads or writes on <code>fd</code> fail unless the original file descriptor allows the operations. For convenience in referring to standard input, standard output, and standard error, an additional set of names is provided: <code>/dev/stdin</code> is a synonym for <code>/dev/fd/0</code>, <code>/dev/stdout</code> for <code>/dev/fd/1</code>, and <code>/dev/stderr</code> for <code>/dev/fd/2</code>.</p>
<b>SEE ALSO</b>	<code>creat(2)</code> , <code>dup(2)</code> , <code>open(2)</code>
<b>DIAGNOSTICS</b>	<code>open(2)</code> returns <code>-1</code> and <code>EBADF</code> if the associated file descriptor is not open.

<b>NAME</b>	filehdr – file header for common object files
<b>SYNOPSIS</b>	<b>#include &lt;filehdr.h&gt;</b>
<b>DESCRIPTION</b>	<p>Every common object file begins with a 20-byte header. The following C <b>struct</b> declaration is used:</p> <pre> <b>struct filehdr</b> <b>{</b>     <b>unsigned short f_magic ; /* magic number */</b>     <b>unsigned short f_nscns ; /* number of sections */</b>     <b>long          f_timdat ; /* time &amp; date stamp */</b>     <b>long          f_sympr ; /* file ptr to symtab */</b>     <b>long          f_nsyms ; /* number of symtab entries */</b>     <b>unsigned short f_opthdr ; /* sizeof(opt and header) */</b>     <b>unsigned short f_flags ; /* flags */</b> <b>};</b> </pre> <p><b>f_sympr</b> is the byte offset into the file at which the symbol table can be found. Its value can be used as the offset in <b>fseek(3S)</b> to position an I/O stream to the symbol table. The UNIX system optional header is 28 bytes. The valid magic numbers are given below:</p> <pre> <b>#define I386MAGIC    0514 /* i386 Computer */</b> <b>#define WE32MAGIC   0560 /* 3B2, 3B5, and 3B15 computers */</b> <b>#define N3BMAGIC    0550 /* 3B20 computer */</b> <b>#define NTVMAGIC    0551 /* 3B20 computer */</b>  <b>#define VAXWRMAGIC  0570 /* VAX writable text segments */</b> <b>#define VAXROMAGIC  0575 /* VAX read only sharable</b> <b>                        text segments */</b> </pre> <p>The value in <b>f_timdat</b> is obtained from the <b>time(2)</b> system call. Flag bits currently defined are:</p> <pre> <b>#define F_RELFLG    0000001 /* relocation entries stripped */</b> <b>#define F_EXEC      0000002 /* file is executable */</b> <b>#define F_LNNO      0000004 /* line numbers stripped */</b> <b>#define F_LSYMS     0000010 /* local symbols stripped */</b> <b>#define F_AR16WR    0000200 /* 16-bit DEC host */</b> <b>#define F_AR32WR    0000400 /* 32-bit DEC host */</b> <b>#define F_AR32W     0001000 /* non-DEC host */</b> <b>#define F_BM32ID    0160000 /* WE32000 family ID field */</b> </pre>

```
#define F_BM32B      0020000 /* file contains WE 32100 code */
#define F_BM32MAU   0040000 /* file reqs MAU to execute */
#define F_BM32RST   0010000 /* this object file contains restore
                             work around [3B5/3B2 only] */
```

**SEE ALSO** [time\(2\)](#), [fseek\(3S\)](#), [a.out\(4\)](#)

<b>NAME</b>	format.dat – disk drive configuration for the format command
<b>DESCRIPTION</b>	<p><b>format.dat</b> enables you to use your specific disk drives with <b>format(1M)</b>. On Solaris 2.3 and later systems, <b>format</b> will automatically configure and label SCSI drives, so that they need not be defined in <b>format.dat</b>. Three things can be defined in the data file:</p> <ul style="list-style-type: none"> <li>• search paths</li> <li>• disk types</li> <li>• partition tables.</li> </ul>
<b>Syntax</b>	<p>The following syntax rules apply to the data file:</p> <ul style="list-style-type: none"> <li>• The pound # sign is the comment character. Any text on a line after a pound sign is not interpreted by <b>format</b>.</li> <li>• Each definition in the <b>format.dat</b> file appears on a single logical line. If the definition is more than one line long, all but the last line of the definition must end with a backslash (\).</li> <li>• A definition consists of a series of assignments that have an identifier on the left side and one or more values on the right side. The assignment operator is the equal sign (=). Assignments within a definition must be separated by a colon (:).</li> <li>• White space is ignored by <b>format(1M)</b>. If you want an assigned value to contain white space, enclose the entire value in double quotes ("). This will cause the white space within quotes to be preserved as part of the assignment value.</li> <li>• Some assignments can have multiple values on the right hand side. Separate values by a comma (,).</li> </ul>
<b>Keywords</b>	<p>The data file contains disk definitions that are read in by <b>format(1M)</b> when it starts up. Each definition starts with one of the following keywords: <b>search_path</b>, <b>disk_type</b>, and <b>partition</b>.</p> <p><b>search_path</b>      4.x: Tells <b>format</b> which disks it should search for when it starts up. The list in the default data file contains all the disks in the GENERIC configuration file. If your system has disks that are not in the GENERIC configuration file, add them to the <b>search_path</b> definition in your data file. The data file can contain only one <b>search_path</b> definition. However, this single definition lets you specify all the disks you have in your system.</p> <p>                         5.x: By default, <b>format(1M)</b> understands all the logical devices that are of the form <b>/dev/rdisk/cntndnsn</b>; hence <b>search_path</b> is not normally defined on a 5.x system.</p> <p><b>disk_type</b>          Defines the controller and disk model. Each <b>disk_type</b> definition contains information concerning the physical geometry of the disk. The default data file contains definitions for the controllers and disks that the Solaris operating system supports. You need to add a new <b>disk_type</b> only if you have an unsupported disk. You can add as many <b>disk_type</b></p>

definitions to the data file as you want.

The following controller types are supported by **format(1M)**:

<b>XY450</b>	Xylogics 450 controller (SMD)
<b>XD7053</b>	Xylogics 7053 controller (SMD)
<b>MD21</b>	SCSI, but using ESDI devices (also known as shoebox)
<b>SCSI</b>	True SCSI (CCS or SCSI-2)
<b>ISP-80</b>	IPI panther controller

Note: The **disk\_type** and **partition** definition entries must have “**ctlr = MD21**” for scsi disk devices for 4.1.1 release. But for 4.1.2, 4.1.3 and 5.x releases, the entries should say “**ctlr=SCSI.**”

The keyword itself is assigned the name of the disk type. This name appears in the disk’s label and is used to identify the disk type whenever **format(1M)** is run. Enclose the name in double quotes to preserve any white space in the name.

Below are lists of identifiers for supported controllers. Note that an asterisk (\*) indicates the identifier is mandatory for that controller -- it is not part of the keyword name.

The following identifiers are assigned values in all **disk\_type** definitions:

<b>acyl*</b>	alternate cylinders
<b>asect</b>	alternate sectors per track
<b>atrks</b>	alternate tracks
<b>fmt_time</b>	formatting time per cylinder
<b>ncyl*</b>	number of logical cylinders
<b>nhead*</b>	number of logical heads
<b>nsect*</b>	number of logical sectors per track
<b>pcyl*</b>	number of physical cylinders
<b>phead</b>	number of physical heads
<b>psect</b>	number of physical sectors per track
<b>rpm*</b>	drive RPM

These identifiers are for SCSI and MD-21 Controllers

<b>read_retries</b>	page 1 byte 3 (read retries)
<b>write_retries</b>	page 1 byte 8 (write retries)
<b>cyl_skew</b>	page 3 bytes 18-19 (cylinder skew)
<b>trk_skew</b>	page 3 bytes 16-17 (track skew)
<b>trks_zone</b>	page 3 bytes 2-3 (tracks per zone)
<b>cache</b>	page 38 byte 2 (cache parameter)
<b>prefetch</b>	page 38 byte 3 (prefetch parameter)
<b>max_prefetch</b>	page 38 byte 4 (minimum prefetch)
<b>min_prefetch</b>	page 38 byte 6 (maximum prefetch)

Note: The Page 38 values are device-specific. Refer the user to the particular disk’s manual for these values.



For SCSI disks, the following geometry specifiers may cause a mode select on the byte(s) indicated:

**asect**           page 3 bytes 4-5 (alternate sectors per zone)  
**atrks**           page 3 bytes 8-9 (alt. tracks per logical unit)  
**phead**           page 4 byte 5 (number of heads)  
**psect**           page 3 bytes 10-11 (sectors per track)

And these identifiers are for SMD Controllers Only

**bps\***            bytes per sector (SMD)  
**bpt\***            bytes per track (SMD)

Note: under SunOS 5.x, bpt is only required for SMD disks. Under SunOS 4.x, bpt was required for all disk types, even though it was only used for SMD disks.

And this identifier is for XY450 SMD Controllers Only

**drive\_type\***    drive type (SMD) (just call this "xy450 drive type")

## partition

Defines a partition table for a specific disk type. The partition table contains the partitioning information, plus a name that lets you refer to it in **format(1M)**. The default data file contains default partition definitions for several kinds of disk drives. Add a partition definition if you repartitioned any of the disks on your system. Add as many partition definitions to the data file as you need.

Partition naming conventions differ in SunOS 4.x and in SunOS 5.x.

4.x: the partitions are named as **a, b, c, d, e, f, g, h**.

5.x: the partitions are referred to by numbers **0, 1, 2, 3, 4, 5, 6, 7**.

## EXAMPLES

Following is a sample **disk\_type** and **partition** definition in **format.dat** file for SUN0535 disk device.

```
disk_type = "SUN0535" \
: ctrl = SCSI : fmt_time = 4 \
: ncyl = 1866 : acyl = 2 : pcyl = 2500 : nhead = 7 : nsect = 80 \
: rpm = 5400

partition = "SUN0535" \
: disk = "SUN0535" : ctrl = SCSI \
: 0 = 0, 64400 : 1 = 115, 103600 : 2 = 0, 1044960 : 6 = 300, 876960
```

## FILES

**/etc/format.dat**

default data file if **format -x** is not specified, nor is there a **format.dat** file in the current directory.

## SEE ALSO

**format(1M)**  
*System Administration Guide*

<b>NAME</b>	fspec – format specification in text files
<b>DESCRIPTION</b>	<p>It is sometimes convenient to maintain text files on the system with non-standard tabs, (tabs that are not set at every eighth column). Such files must generally be converted to a standard format, frequently by replacing all tabs with the appropriate number of spaces, before they can be processed by system commands. A format specification occurring in the first line of a text file specifies how tabs are to be expanded in the remainder of the file.</p> <p>A format specification consists of a sequence of parameters separated by blanks and surrounded by the brackets &lt;: and &gt;:. Each parameter consists of a keyletter, possibly followed immediately by a value. The following parameters are recognized:</p> <p><b>ttabs</b>     The <b>t</b> parameter specifies the tab settings for the file. The value of <b>tabs</b> must be one of the following:</p> <ul style="list-style-type: none"> <li>A list of column numbers separated by commas, indicating tabs set at the specified columns</li> <li>A '-' followed immediately by an integer <i>n</i>, indicating tabs at intervals of <i>n</i> columns</li> <li>A '-' followed by the name of a "canned" tab specification</li> </ul> <p>Standard tabs are specified by <b>t-8</b>, or equivalently, <b>t1,9,17,25</b>, etc. The canned tabs that are recognized are defined by the <b>tabs(1)</b> command.</p> <p><b>ssize</b>     The <b>s</b> parameter specifies a maximum line size. The value of <i>size</i> must be an integer. Size checking is performed after tabs have been expanded, but before the margin is prepended.</p> <p><b>mmargin</b>   The <b>m</b> parameter specifies a number of spaces to be prepended to each line. The value of <i>margin</i> must be an integer.</p> <p><b>d</b>           The <b>d</b> parameter takes no value. Its presence indicates that the line containing the format specification is to be deleted from the converted file.</p> <p><b>e</b>           The <b>e</b> parameter takes no value. Its presence indicates that the current format is to prevail only until another format specification is encountered in the file.</p> <p>Default values, which are assumed for parameters not supplied, are <b>t-8</b> and <b>m0</b>. If the <b>s</b> parameter is not specified, no size checking is performed. If the first line of a file does not contain a format specification, the above defaults are assumed for the entire file. The following is an example of a line containing a format specification:</p> <p style="text-align: center;">* &lt;:t5,10,15 s72:&gt; *</p> <p>If a format specification can be disguised as a comment, it is not necessary to code the <b>d</b> parameter.</p>
<b>SEE ALSO</b>	<b>ed(1)</b> , <b>newform(1)</b> , <b>tabs(1)</b>

<b>NAME</b>	<b>fstypes</b> – file that registers distributed file system packages
<b>DESCRIPTION</b>	<p><b>fstypes</b> resides in directory <b>/etc/dfs</b> and lists distributed file system utilities packages installed on the system. For each installed distributed file system type, there is a line that begins with the file system type name (for example, “nfs”), followed by white space and descriptive text.</p> <p>The file system indicated in the first line of the file is the default file system; when Distributed File System (DFS) Administration commands are entered without the option <b>-F <i>fstypes</i></b>, the system takes the file system type from the first line of the <b>fstypes</b> file.</p> <p>The default file system can be changed by editing the <b>fstypes</b> file with any supported text editor.</p>
<b>SEE ALSO</b>	<b>dfmounts(1M)</b> , <b>dfshares(1M)</b> , <b>share(1M)</b> , <b>shareall(1M)</b> , <b>unshare(1M)</b>

<b>NAME</b>	fs_ufs, inode_ufs, inode – format of a ufs file system volume
<b>SYNOPSIS</b>	<pre>#include &lt;sys/param.h&gt; #include &lt;sys/types.h&gt; #include &lt;sys/fs/ufs_fs.h&gt; #include &lt;sys/fs/ufs_inode.h&gt;</pre>
<b>DESCRIPTION</b>	<p>Standard UFS file system storage volumes have a common format for certain vital information. Every volume is divided into a certain number of blocks. The block size is a parameter of the file system. Sectors 0 to 15 contain primary and secondary bootstrapping programs.</p> <p>The actual file system begins at sector 16 with the super-block. The layout of the super-block is defined by the header <code>&lt;sys/fs/ufs_fs.h&gt;</code>.</p> <p>Each disk drive contains some number of file systems. A file system consists of a number of cylinder groups. Each cylinder group has inodes and data.</p> <p>A file system is described by its super-block, and by the information in the cylinder group blocks. The super-block is critical data and is replicated before each cylinder group block to protect against catastrophic loss. This is done at file system creation time and the critical super-block data does not change, so the copies need not be referenced.</p>
<b>fs_clean</b>	<p><b>fs_clean</b> indicates the state of the file system. The <b>FSCLEAN</b> state indicates an undamaged, cleanly unmounted file system. The <b>FSACTIVE</b> state indicates a mounted file system that has been updated. The <b>FSSTABLE</b> state indicates an idle mounted file system. The <b>FSFIX</b> state indicates that this fs is mounted, contains inconsistent file system data and is being repaired by <b>fsck</b>. The <b>FSBAD</b> state indicates that this file system contains inconsistent file system data. It is not necessary to run <b>fsck</b> on any unmounted file systems with a state of <b>FSCLEAN</b> or <b>FSSTABLE</b>. <b>mount(2)</b> will return <b>ENOSPC</b> if a UFS file system with a state of <b>FSACTIVE</b> is being mounted for read-write.</p> <p>To provide additional safeguard, <b>fs_clean</b> could be trusted only if <b>fs_state</b> contains a value equal to <b>FSOKAY - fs_time</b>, where <b>FSOKAY</b> is a constant integer. Otherwise, <b>fs_clean</b> is treated as though it contains the value of <b>FSACTIVE</b>.</p> <p>Addresses stored in inodes are capable of addressing fragments of “blocks.” File system blocks of at most, size <b>MAXBSIZE</b> can be optionally broken into 2, 4, or 8 pieces, each of which is addressable; these pieces may be <b>DEV_BSIZE</b> or some multiple of a <b>DEV_BSIZE</b> unit.</p> <p>Large files consist exclusively of large data blocks. To avoid undue wasted disk space, the last data block of a small file is allocated only as many fragments of a large block as are necessary. The file system format retains only a single pointer to such a fragment, which is a piece of a single large block that has been divided. The size of such a fragment is determinable from information in the inode, using the <b>blksize(fs, ip, lbn)</b> macro.</p>

	<p>The file system records space availability at the fragment level; aligned fragments are examined to determine block availability.</p> <p>The root inode is the root of the file system. Inode 0 cannot be used for normal purposes and historically, bad blocks were linked to inode 1. Thus the root inode is 2 (inode 1 is no longer used for this purpose; however numerous dump tapes make this assumption, so we are stuck with it). The <i>lost+found</i> directory is given the next available inode when it is initially created by <b>mkfs(1M)</b>.</p>
<b>fs_minfree</b>	<p><b>fs_minfree</b> gives the minimum acceptable percentage of file system blocks which may be free. If the freelist drops below this level only the super-user may continue to allocate blocks. <b>fs_minfree</b> may be set to 0 if no reserve of free blocks is deemed necessary, however severe performance degradations will be observed if the file system is run at greater than 90% full; thus the default value of <b>fs_minfree</b> is 10%.</p> <p>Empirically the best trade-off between block fragmentation and overall disk utilization at a loading of 90% comes with a fragmentation of 8; thus the default fragment size is an eighth of the block size.</p>
<b>fs_optim</b>	<p><b>fs_optim</b> specifies whether the file system should try to minimize the time spent allocating blocks, or if it should attempt to minimize the space fragmentation on the disk. If the value of <b>fs_minfree</b> is less than 10%, then the file system defaults to optimizing for space to avoid running out of full sized blocks. If the value of <b>fs_minfree</b> is greater than or equal to 10%, fragmentation is unlikely to be problematical, and the file system defaults to optimizing for time.</p> <p><i>Cylinder group related limits:</i> Each cylinder keeps track of the availability of blocks at different rotational positions, so that sequential blocks can be laid out with minimum rotational latency. <b>fs_nrpos</b> is the number of rotational positions which are distinguished. With the default <b>fs_nrpos</b> of 8, the resolution of the summary information is 2ms for a typical 3600 rpm drive.</p>
<b>fs_rotdelay</b>	<p><b>fs_rotdelay</b> gives the minimum number of milliseconds to initiate another disk transfer on the same cylinder. It is used in determining the rotationally optimal layout for disk blocks within a file; the default value for <b>fs_rotdelay</b> varies from drive to drive (see <b>tunefs(1M)</b>).</p>
<b>fs_maxcontig</b>	<p><b>fs_maxcontig</b> gives the maximum number of blocks, belonging to one file, that will be allocated contiguously before inserting a rotational delay.</p> <p>Each file system has a statically allocated number of inodes. An inode is allocated for each NBPI bytes of disk space. The inode allocation strategy is extremely conservative. <b>MINBSIZE</b> is the smallest allowable block size. With a <b>MINBSIZE</b> of 4096 it is possible to create files of size <math>2^{32}</math> with only two levels of indirection. <b>MINBSIZE</b> must be large enough to hold a cylinder group block, thus changes to <b>(struct cg)</b> must keep its size within <b>MINBSIZE</b>. Note: super-blocks are never more than size <b>SBSIZE</b>.</p>

The path name on which the file system is mounted is maintained in **fs\_fsmnt**. **MAXMNTLEN** defines the amount of space allocated in the super-block for this name.

The limit on the amount of summary information per file system is defined by **MAXCSBUFS**. It is currently parameterized for a maximum of two million cylinders.

Per cylinder group information is summarized in blocks allocated from the first cylinder group's data blocks. These blocks are read in from **fs\_csaddr** (size **fs\_cssize**) in addition to the super-block.

Note: **sizeof (struct csum)** must be a power of two in order for the **fs\_cs** macro to work.

The inode is the focus of all file activity in the file system. There is a unique inode allocated for each active file, each current directory, each mounted-on file, text file, and the root. An inode is "named" by its device/i-number pair. For further information, see the header **<sys/fs/ufs\_inode.h>**.

**ATTRIBUTES**

See **attributes(5)** for a description of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Stability Level	Unstable

**SEE ALSO**

**fsck\_ufs(1M)**, **mkfs\_ufs(1M)**, **tunefs(1M)**, **mount(2)**, **attributes(5)**

<b>NAME</b>	group – group file
<b>DESCRIPTION</b>	<p>The <b>group</b> file is a local source of group information. The <b>group</b> file can be used in conjunction with other group sources, including the NIS maps <b>group.byname</b> and <b>group.bygid</b> and the NIS+ table <b>group</b>. Programs use the <b>getgrnam(3C)</b> routines to access this information.</p> <p>The <b>group</b> file contains a one-line entry for each group recognized by the system, of the form:</p> <p style="padding-left: 40px;"><i>groupname:password:gid:user-list</i></p> <p>where</p> <p style="padding-left: 40px;"><i>groupname</i>    The name of the group.</p> <p style="padding-left: 40px;"><i>gid</i>            The group's unique numerical ID (GID) within the system.</p> <p style="padding-left: 40px;"><i>user-list</i>     A comma-separated list of users allowed in the group.</p> <p>The maximum value of the <i>gid</i> field is 2137483647. To maximize interoperability and compatibility, administrators are recommended to assign groups using the range of GIDs below 60000 where possible.</p> <p>If the password field is empty, no password is demanded. During user identification and authentication, the supplementary group access list is initialized sequentially from information in this file. If a user is in more groups than the system is configured for, {NGROUPS_MAX}, a warning will be given and subsequent group specifications will be ignored.</p> <p>Malformed entries cause routines that read this file to halt, in which case group assignments specified further along are never made. To prevent this from happening, use <b>grpck(1B)</b> to check the <b>/etc/group</b> database from time to time.</p> <p>Previous releases used a group entry beginning with a '+' (plus sign) or '-' (minus sign) to selectively incorporate entries from NIS maps for group. If still required, this is supported by specifying <b>group:compat</b> in <b>nsswitch.conf(4)</b>. The "compat" source may not be supported in future releases. The preferred sources are, "files" followed by "nisplus". This has the effect of incorporating the entire contents of the NIS+ <b>group</b> table after the <b>group</b> file.</p>
<b>EXAMPLES</b>	<p>Here is a sample <b>group</b> file:</p> <p style="padding-left: 40px;"><b>root::0:root</b></p> <p style="padding-left: 40px;"><b>stooges:q.mJzTnu8icF.:10:larry,moe,curly</b></p> <p>and the sample group entry from <b>nsswitch.conf</b>:</p> <p style="padding-left: 40px;"><b>group: files nisplus</b></p> <p>With these entries, the group <b>stooges</b> will have members <b>larry</b>, <b>moe</b>, and <b>curly</b>, and all groups listed in the NIS+ group table are effectively incorporated after the entry for <b>stooges</b>.</p>

If the **group** file was:

```
root::0:root
```

```
stooges:q.mJzTnu8icF.:10:larry,moe,curly
```

```
+:
```

and the group entry from `nsswitch.conf`:

```
group: compat
```

all the groups listed in the NIS **group.bygid** and **group.byname** maps would be effectively incorporated after the entry for `stooges`.

**SEE ALSO**

**groups(1)**, **grpck(1B)**, **newgrp(1)**, **getgrnam(3C)**, **initgroups(3C)**, **nsswitch.conf(4)**, **unistd(4)**

*System Administration Guide*



<b>NAME</b>	holidays – prime/nonprime table for the accounting system
<b>SYNOPSIS</b>	<i>/etc/acct/holidays</i>
<b>DESCRIPTION</b>	<p>The <i>/etc/acct/holidays</i> file describes which hours are considered prime time and which days are holidays. Holidays and weekends are considered non-prime time hours. <i>/etc/acct/holidays</i> is used by the accounting system.</p> <p>All lines beginning with an "*" are comments.</p> <p>The <i>/etc/acct/holidays</i> file consists of two sections. The first non-comment line defines the current year and the start time of prime and non-prime time hours, in the form:</p> <pre style="margin-left: 40px;"><i>current_year prime_start non_prime_start</i></pre> <p>The remaining non-comment lines define the holidays in the form:</p> <pre style="margin-left: 40px;"><i>month/day company_holiday</i></pre> <p>Of these two fields, only the <i>month/day</i> is actually used by the accounting system programs.</p> <p>The <i>/etc/acct/holidays</i> file must be updated each year.</p>
<b>EXAMPLES</b>	<p>The following is an example of the <i>/etc/acct/holidays</i> file:</p> <pre style="margin-left: 40px;">* <b>Prime/Nonprime Table for the accounting system</b> * * <b>Curr Prime Non-Prime</b> * <b>Year Start Start</b> * * <b>1991 0830 1800</b> * * <b>only the first column (month/day) is significant.</b> * * <b>month/day Company</b> * <b>          Holiday</b> * * <b>1/1          New Years Day</b> * <b>5/30          Memorial Day</b> * <b>7/4          Indep. Day</b> * <b>9/5          Labor Day</b> * <b>11/24        Thanksgiving Day</b> * <b>11/25        day after Thanksgiving</b> * <b>12/25        Christmas</b> * <b>12/26        day after Christmas</b></pre>
<b>SEE ALSO</b>	<i>acct(1M)</i>

<b>NAME</b>	hosts – host name database
<b>SYNOPSIS</b>	<i>/etc/inet/hosts</i> <i>/etc/hosts</i>
<b>DESCRIPTION</b>	<p>The <b>hosts</b> file is a local database that associates the names of hosts with their Internet Protocol (IP) addresses. The <b>hosts</b> file can be used in conjunction with, or instead of, other hosts databases, including the Domain Name System (DNS), the NIS <b>hosts</b> map and the NIS+ <b>hosts</b> table. Programs use library interfaces to access information in the <b>hosts</b> file.</p> <p>The <b>hosts</b> file has one entry for each IP address of each host. If a host has more than one IP address, it will have one entry for each, on consecutive lines. The format of each line is:</p> <pre style="margin-left: 40px;"><i>IP-address</i>      <i>official-host-name</i>      <i>nicknames...</i></pre> <p>Items are separated by any number of SPACE and/or TAB characters. The first item on a line is the host's IP address. The second entry is the host's official name. Subsequent entries on the same line are alternative names for the same machine, or "nicknames." Nicknames are optional.</p> <p>For a host with more than one IP address, consecutive entries for these addresses may contain the same or differing nicknames. Different nicknames are useful for assigning distinct names to different addresses.</p> <p>A call to <b>gethostbyname</b>(3N) returns a <b>hostent</b> structure containing the union of all addresses and nicknames from each line containing a matching official name or nickname.</p> <p>A '#' indicates the beginning of a comment; characters up to the end of the line are not interpreted by routines that search the file.</p> <p>Network addresses are written in the conventional "decimal dot" notation and interpreted using the <b>inet_addr</b> routine from the Internet address manipulation library, <b>inet</b>(3N).</p> <p>This interface supports host names as defined in Internet RFC 952 which states:</p> <p>A "name" (Net, Host, Gateway, or Domain name) is a text string up to 24 characters drawn from the alphabet (A-Z), digits (0-9), minus sign (-), and period (.). Note that periods are only allowed when they serve to delimit components of "domain style names". (See RFC 921, "Domain Name System Implementation Schedule," for background). No blank or space characters are permitted as part of a name. No distinction is made between upper and lower case. The first character must be an alpha character. The last character must not be a minus sign or period.</p> <p>Although the interface accepts host names longer than 24 characters for the host portion (exclusive of the domain component), choosing names for hosts that adhere to the 24 character restriction will insure maximum interoperability on the Internet.</p>

A host which serves as a GATEWAY should have “-GATEWAY” or “-GW” as part of its name. Hosts which do not serve as Internet gateways should not use “-GATEWAY” and “-GW” as part of their names. A host which is a TAC should have “-TAC” as the last part of its host name, if it is a DoD host. Single character names or nicknames are not allowed.

RFC 952 has been modified by RFC 1123 to relax the restriction on the first character being a digit.

**EXAMPLES**

Here is a typical line from the **hosts** file:

```
192.9.1.20    gaia           # John Smith
```

**SEE ALSO**

**in.named(1M)**, **gethostbyname(3N)**, **inet(3N)**, **nsswitch.conf(4)**, **resolv.conf(4)**

**NOTES**

**/etc/inet/hosts** is the official SVR4 name of the **hosts** file. The symbolic link **/etc/hosts** exists for BSD compatibility.

<b>NAME</b>	hosts.equiv, rhosts – trusted remote hosts and users
<b>DESCRIPTION</b>	<p>The <b>/etc/hosts.equiv</b> and <b>.rhosts</b> files provide the “remote authentication” database for <b>rlogin(1)</b>, <b>rsh(1)</b>, <b>rcp(1)</b>, and <b>rcmd(3N)</b>. The files specify remote hosts and users that are considered <i>trusted</i>. Trusted users are allowed to access the local system <i>without supplying a password</i>. The library routine <b>ruserok()</b> (see <b>rcmd(3N)</b>) performs the authentication procedure for programs by using the <b>/etc/hosts.equiv</b> and <b>.rhosts</b> files. The <b>/etc/hosts.equiv</b> file applies to the entire system, while individual users can maintain their own <b>.rhosts</b> files in their home directories.</p> <p>These files <i>bypass</i> the standard password-based user authentication mechanism. To maintain system security, care must be taken in creating and maintaining these files.</p> <p>The remote authentication procedure determines whether a user from a remote host should be allowed to access the local system with the identity of a local user. This procedure first checks the <b>/etc/hosts.equiv</b> file and then checks the <b>.rhosts</b> file in the home directory of the local user who is requesting access. Entries in these files can be of two forms. <i>Positive</i> entries <i>allow</i> access, while <i>negative</i> entries <i>deny</i> access. The authentication succeeds when a matching positive entry is found. The procedure fails when the first matching negative entry is found, or if no matching entries are found in either file. Thus, the order of entries is important; If the files contain positive and negative entries, the entry that appears first will prevail. The <b>rsh(1)</b> and <b>rcp(1)</b> programs fail if the remote authentication procedure fails. The <b>rlogin</b> program falls back to the standard password-based login procedure if the remote authentication fails.</p> <p>Both files are formatted as a list of one-line entries. Each entry has the form:</p> <p style="padding-left: 40px;"><i>hostname [username]</i></p> <p>Negative entries are differentiated from positive entries by a ‘-’ character preceding either the <i>hostname</i> or <i>username</i> field.</p> <p><b>Positive Entries</b></p> <p>If the form:</p> <p style="padding-left: 40px;"><i>hostname</i></p> <p>is used, then users from the named host are trusted. That is, they may access the system with the same user name as they have on the remote system. This form may be used in both the <b>/etc/hosts.equiv</b> and <b>.rhosts</b> files.</p> <p>If the line is in the form:</p> <p style="padding-left: 40px;"><i>hostname username</i></p> <p>then the named user from the named host can access the system. This form may be used in individual <b>.rhosts</b> files to allow remote users to access the system <i>as a different local user</i>. If this form is used in the <b>/etc/hosts.equiv</b> file, the named remote user will be allowed to access the system as <i>any</i> local user.</p> <p><b>netgroup(4)</b> can be used in either the <i>hostname</i> or <i>username</i> fields to match a number of hosts or users in one entry. The form:</p> <p style="padding-left: 40px;"><i>+@netgroup</i></p>

allows access from all hosts in the named netgroup. When used in the *username* field, netgroups allow a group of remote users to access the system as a particular local user. The form:

*hostname* +@*netgroup*

allows all of the users in the named netgroup from the named host to access the system as the local user. The form:

+@*netgroup1* +@*netgroup2*

allows the users in *netgroup2* from the hosts in *netgroup1* to access the system as the local user.

The special character '+' can be used in place of either *hostname* or *username* to match any host or user. For example, the entry

+

will allow a user from any remote host to access the system with the same username. The entry

+ *username*

will allow the named user from any remote host to access the system. The entry

*hostname* +

will allow any user from the named host to access the system as the local user.

#### Negative Entries

Negative entries are preceded by a '/'-' sign. The form:

-*hostname*

will disallow all access from the named host. The form:

-@*netgroup*

means that access is explicitly disallowed from all hosts in the named netgroup. The form:

*hostname* -*username*

disallows access by the named user only from the named host, while the form:

+ -@*netgroup*

will disallow access by all of the users in the named netgroup from all hosts.

#### FILES

**/etc/hosts.equiv**  
**~/rhosts**

#### SEE ALSO

**rcp(1)**, **rlogin(1)**, **rsh(1)**, **rcmd(3N)**, **hosts(4)**, **netgroup(4)**, **passwd(4)**

#### NOTES

Hostnames in **/etc/hosts.equiv** and **.rhosts** files must be the official name of the host, not one of its nicknames.

Root access is handled as a special case. Only the **.rhosts** file is checked when access is being attempted for root. To help maintain system security, the **/etc/hosts.equiv** file is not checked.

As a security feature, the **.rhosts** file must be owned by the user who is attempting access. Positive entries in **/etc/hosts.equiv** that include a *username* field (either an individual named user, a netgroup, or '+' sign) should be used with extreme caution. Because **/etc/hosts.equiv** applies system-wide, these entries allow one, or a group of, remote users to access the system as *any local user*. This can be a security hole.

<b>NAME</b>	inetd.conf – Internet servers database										
<b>SYNOPSIS</b>	<b>/etc/inet/inetd.conf</b> <b>/etc/inetd.conf</b>										
<b>DESCRIPTION</b>	<p>The <b>inetd.conf</b> file contains the list of servers that <b>inetd</b>(1M) invokes when it receives an Internet request over a socket. Each server entry is composed of a single line of the form:</p> <p><i>service-name endpoint-type protocol wait-status uid server-program server-arguments</i></p> <p>Fields are separated by either SPACE or TAB characters. A '#' (number sign) indicates the beginning of a comment; characters up to the end of the line are not interpreted by routines that search this file.</p> <p><i>service-name</i>            The name of a valid service listed in the <b>services</b> file. For RPC services, the value of the <i>service-name</i> field consists of the RPC service name or program number, followed by a '/' (slash) and either a version number or a range of version numbers (for example, <b>rstatd/2-4</b>).</p> <p><i>endpoint-type</i>           Can be one of:</p> <table border="0" style="margin-left: 40px;"> <tr> <td><b>stream</b></td> <td>for a stream socket,</td> </tr> <tr> <td><b>dgram</b></td> <td>for a datagram socket,</td> </tr> <tr> <td><b>raw</b></td> <td>for a raw socket,</td> </tr> <tr> <td><b>seqpacket</b></td> <td>for a sequenced packet socket</td> </tr> <tr> <td><b>tli</b></td> <td>for all tli endpoints</td> </tr> </table> <p><i>protocol</i>                Must be a recognized protocol listed in the file <b>/etc/inet/protocols</b>. For RPC services, the field consists of the string <b>rpc</b> followed by a '/' (slash) and either a '*' (asterisk), one or more nettypes, one or more netids, or a combination of nettypes and netids. Whatever the value, it is first treated as a nettype. If it is not a valid nettype, then it is treated as a netid. For example, <b>rpc/*</b> for an RPC service using all the transports supported by the system (the list can be found in the <b>/etc/netconfig</b> file), equivalent to saying <b>rpc/visible rpc/ticots</b> for an RPC service using the Connection-Oriented Transport Service.</p> <p><i>wait-status</i>            <b>nowait</b> for all but "single-threaded" datagram servers — servers which do not release the socket until a timeout occurs. These must have the status <b>wait</b>. Do not configure <b>udp</b> services as <b>nowait</b>. This will cause a race condition where the <b>inetd</b> 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.</p> <p><i>uid</i>                      The user ID under which the server should run. This allows servers to run with access privileges other than those for root.</p> <p><i>server-program</i>        Either the pathname of a server program to be invoked by <b>inetd</b> to</p>	<b>stream</b>	for a stream socket,	<b>dgram</b>	for a datagram socket,	<b>raw</b>	for a raw socket,	<b>seqpacket</b>	for a sequenced packet socket	<b>tli</b>	for all tli endpoints
<b>stream</b>	for a stream socket,										
<b>dgram</b>	for a datagram socket,										
<b>raw</b>	for a raw socket,										
<b>seqpacket</b>	for a sequenced packet socket										
<b>tli</b>	for all tli endpoints										

perform the requested service, or the value **internal** if **inetd** itself provides the service.

*server-arguments*

If a server must be invoked with command line arguments, the entire command line (including argument 0) must appear in this field (which consists of all remaining words in the entry). If the server expects **inetd** to pass it the address of its peer (for compatibility with 4.2BSD executable daemons), then the first argument to the command should be specified as '%A'. No more than five arguments are allowed in this field.

**FILES**

**/etc/netconfig** network configuration file  
**/etc/inet/protocols** Internet protocols  
**/etc/inet/services** Internet network services

**SEE ALSO**

**rlogin(1)**, **rsh(1)**, **in.tftpd(1M)**, **inetd(1M)**, **services(4)**

**NOTES**

**/etc/inet/inetd.conf** is the official SVR4 name of the **inetd.conf** file. The symbolic link **/etc/inetd.conf** exists for BSD compatibility.



<b>NAME</b>	init.d – initialization and termination scripts for changing init states
<b>SYNOPSIS</b>	<code>/etc/init.d</code>
<b>DESCRIPTION</b>	<p><code>/etc/init.d</code> is a directory containing initialization and termination scripts for changing init states. These scripts are linked when appropriate to files in the <code>rc?.d</code> directories, where '?' is a single character corresponding to the init state. See <code>init(1M)</code> for definitions of the states.</p> <p>File names in <code>rc?.d</code> directories are of the form <code>[SK]nn&lt;init.d filename&gt;</code>, where <b>S</b> means start this job, <b>K</b> means kill this job, and <b>nn</b> is the relative sequence number for killing or starting the job. When entering a state (init S,0,2,3,etc.) the <code>rc[S0-6]</code> script executes those scripts in <code>/etc/rc[S0-6].d</code> that are prefixed with <b>K</b> followed by those scripts prefixed with <b>S</b>. When executing each script in one of the <code>/etc/rc[S0-6]</code> directories, the <code>/sbin/rc[S0-6]</code> script passes a single argument. It passes the argument 'stop' for scripts prefixed with <b>K</b> and the argument 'start' for scripts prefixed with <b>S</b>. There is no harm in applying the same sequence number to multiple scripts. In this case the order of execution is deterministic but unspecified.</p> <p>Guidelines for selecting sequence numbers are provided in <b>README</b> files located in the directory associated with that target state. For example, <code>/etc/rc[S0-6].d/README</code>. Absence of a <b>README</b> file indicates that there are currently no established guidelines.</p>
<b>EXAMPLES</b>	<p>When changing to init state 2 (multi-user mode, network resources not exported), <code>/sbin/rc2</code> is initiated by the init process. The following steps are performed by <code>/sbin/rc2</code>.</p> <ol style="list-style-type: none"> <li>1. In the directory <code>/etc/rc2.d</code> are files used to stop processes that should not be running in state 2. The filenames are prefixed with <b>K</b>. Each <b>K</b> file in the directory is executed (by <code>/sbin/rc2</code>) in alpha-numeric order when the system enters init state 2. See example below.</li> <li>2. Also in the <code>rc2.d</code> directory are files used to start processes that should be running in state 2. As in the Step 1, each <b>S</b> file is executed.</li> </ol> <p>Assume the file <code>/etc/netdaemon</code> is a script that will initiate networking daemons when given the argument 'start', and will terminate the daemons if given the argument 'stop'. It is linked to <code>/etc/rc2.d/S68netdaemon</code>, and to <code>/etc/rc0.d/K67netdaemon</code>. The file is executed by <code>/etc/rc2.d/S68netdaemon start</code> when init state 2 is entered and by <code>/etc/rc0.d/S67netdaemon stop</code> when shutting the system down.</p>
<b>SEE ALSO</b>	<code>init(1M)</code>
<b>NOTES</b>	<code>/sbin/rc2</code> has references to the obsolescent <code>rc.d</code> directory. These references are for compatibility with old <b>INSTALL</b> scripts. New <b>INSTALL</b> scripts should use the <code>init.d</code> directory for related executables. The same is true for the <code>shutdown.d</code> directory.

<b>NAME</b>	inittab – script for init
<b>DESCRIPTION</b>	<p>The file <code>/etc/inittab</code> controls process dispatching by <b>init</b>. The processes most typically dispatched by <b>init</b> are daemons.</p> <p>The <b>inittab</b> file is composed of entries that are position dependent and have the following format:</p> <p style="text-align: center;"><i>id:rstate:action:process</i></p> <p>Each entry is delimited by a newline; however, a backslash (\) preceding a newline indicates a continuation of the entry. Up to 512 characters for each entry are permitted. Comments may be inserted in the <i>process</i> field using the convention for comments described in <b>sh</b>(1). There are no limits (other than maximum entry size) imposed on the number of entries in the <b>inittab</b> file. The entry fields are:</p> <p><i>id</i>        One or two characters used to uniquely identify an entry.</p> <p><i>rstate</i>    Define the run level in which this entry is to be processed. Run-levels effectively correspond to a configuration of processes in the system. That is, each process spawned by <b>init</b> is assigned a run level(s) in which it is allowed to exist. The run levels are represented by a number ranging from <b>0</b> through <b>6</b>. For example, if the system is in run level <b>1</b>, only those entries having a <b>1</b> in the <i>rstate</i> field are processed.</p> <p>When <b>init</b> is requested to change run levels, all processes that do not have an entry in the <i>rstate</i> field for the target run level are sent the warning signal <b>SIGTERM</b> and allowed a 5-second grace period before being forcibly terminated by the kill signal <b>SIGKILL</b>. The <i>rstate</i> field can define multiple run levels for a process by selecting more than one run level in any combination from <b>0</b> through <b>6</b>. If no run level is specified, then the process is assumed to be valid at all run levels <b>0</b> through <b>6</b>.</p> <p>There are three other values, <b>a</b>, <b>b</b> and <b>c</b>, which can appear in the <i>rstate</i> field, even though they are not true run levels. Entries which have these characters in the <i>rstate</i> field are processed only when an <b>init</b> or <b>telinit</b> process requests them to be run (regardless of the current run level of the system). See <b>init</b>(1M). These differ from run levels in that <b>init</b> can never enter run level <b>a</b>, <b>b</b> or <b>c</b>. Also, a request for the execution of any of these processes does not change the current run level. Furthermore, a process started by an <b>a</b>, <b>b</b> or <b>c</b> command is not killed when <b>init</b> changes levels. They are killed only if their line in <b>inittab</b> is marked <b>off</b> in the <i>action</i> field, their line is deleted entirely from <b>inittab</b>, or <b>init</b> goes into single-user state.</p> <p><i>action</i>    Key words in this field tell <b>init</b> how to treat the process specified in the <i>process</i> field. The actions recognized by <b>init</b> are as follows:</p> <p style="padding-left: 20px;"><b>respawn</b>    If the process does not exist, then start the process; do not wait for its termination (continue scanning the <b>inittab</b> file), and when the process dies, restart the process. If the process currently exists, do nothing and continue scanning the <b>inittab</b> file.</p>

<b>wait</b>	When <b>init</b> enters the run level that matches the entry's <i>rstate</i> , start the process and wait for its termination. All subsequent reads of the <b>inittab</b> file while <b>init</b> is in the same run level cause <b>init</b> to ignore this entry.
<b>once</b>	When <b>init</b> enters a run level that matches the entry's <i>rstate</i> , start the process, do not wait for its termination. When it dies, do not restart the process. If <b>init</b> enters a new run level and the process is still running from a previous run level change, the program is not restarted.
<b>boot</b>	The entry is to be processed only at <b>init</b> 's boot-time read of the <b>inittab</b> file. <b>init</b> is to start the process and not wait for its termination; when it dies, it does not restart the process. In order for this instruction to be meaningful, the <i>rstate</i> should be the default or it must match <b>init</b> 's run level at boot time. This action is useful for an initialization function following a hardware reboot of the system.
<b>bootwait</b>	The entry is to be processed the first time <b>init</b> goes from single-user to multi-user state after the system is booted. (If <b>initdefault</b> is set to <b>2</b> , the process runs right after the boot.) <b>init</b> starts the process, waits for its termination and, when it dies, does not restart the process.
<b>powerfail</b>	Execute the process associated with this entry only when <b>init</b> receives a power fail signal, <b>SIGPWR</b> (see <b>signal(3C)</b> ).
<b>powerwait</b>	Execute the process associated with this entry only when <b>init</b> receives a power fail signal, <b>SIGPWR</b> , and wait until it terminates before continuing any processing of <b>inittab</b> .
<b>off</b>	If the process associated with this entry is currently running, send the warning signal <b>SIGTERM</b> and wait 5 seconds before forcibly terminating the process with the kill signal <b>SIGKILL</b> . If the process is nonexistent, ignore the entry.
<b>ondemand</b>	This instruction is really a synonym for the <b>respawn</b> action. It is functionally identical to <b>respawn</b> but is given a different keyword in order to divorce its association with run levels. This instruction is used only with the <b>a</b> , <b>b</b> or <b>c</b> values described in the <i>rstate</i> field.
<b>initdefault</b>	An entry with this action is scanned only when <b>init</b> is initially invoked. <b>init</b> uses this entry to determine which run level to enter initially. It does this by taking the highest run level specified in the <i>rstate</i> field and using that as its initial state. If the <i>rstate</i> field is empty, this is interpreted as <b>0123456</b> and <b>init</b> will enter run level <b>6</b> . This will cause the system to loop (it will go to firmware and reboot continuously). Additionally, if <b>init</b> does not find an <b>initdefault</b> entry in <b>inittab</b> , it requests an initial run level from the user at reboot time.

**sysinit** Entries of this type are executed before **init** tries to access the console (that is, before the **Console Login:** prompt). It is expected that this entry will be used only to initialize devices that **init** might try to ask the run level question. These entries are executed and **init** waits for their completion before continuing.

*process* Specify a command to be executed. The entire **process** field is prefixed with **exec** and passed to a forked **sh** as **sh -c 'exec command'**. For this reason, any legal **sh** syntax can appear in the *process* field.

**SEE ALSO**

**sh(1)**, **who(1)**, **init(1M)**, **ttymon(1M)**, **exec(2)**, **open(2)**, **signal(3C)**

<b>NAME</b>	issue – issue identification file
<b>DESCRIPTION</b>	The file <code>/etc/issue</code> contains the issue or project identification to be printed as a login prompt. <b>issue</b> is an ASCII file that is read by program <b>getty</b> and then written to any terminal spawned or respawned from the <i>lines</i> file.
<b>FILES</b>	<code>/etc/issue</code>
<b>SEE ALSO</b>	<code>login(1)</code>

<b>NAME</b>	keytables – keyboard table descriptions for loadkeys and dumpkeys
<b>DESCRIPTION</b>	<p>These files are used by <b>loadkeys(1)</b> to modify the translation tables used by the keyboard streams module and generated by (see <b>loadkeys(1)</b>) from those translation tables.</p> <p>Any line in the file beginning with # is a comment, and is ignored. # is treated specially only at the beginning of a line.</p> <p>Other lines specify the values to load into the tables for a particular keystation. The format is either:</p> <p style="padding-left: 40px;"><b>key</b> <i>number list_of_entries</i></p> <p>or</p> <p style="padding-left: 40px;"><b>swap</b> <i>number1 with number2</i></p> <p>or</p> <p style="padding-left: 40px;"><b>key</b> <i>number1 same as number2</i></p> <p>or a blank line, which is ignored.</p> <p style="padding-left: 40px;"><b>key</b> <i>number list_of_entries</i></p> <p>sets the entries for keystation <i>number</i> from the list given. An entry in that list is of the form</p> <p style="padding-left: 40px;"><i>tablename code</i></p> <p>where <i>tablename</i> is the name of a particular translation table, or <b>all</b>. The translation tables are:</p> <p style="padding-left: 40px;"><b>base</b> entry when no shifts are active</p> <p style="padding-left: 40px;"><b>shift</b> entry when "Shift" key is down</p> <p style="padding-left: 40px;"><b>caps</b> entry when "Caps Lock" is in effect</p> <p style="padding-left: 40px;"><b>ctrl</b> entry when "Control" is down</p> <p style="padding-left: 40px;"><b>altg</b> entry when "Alt Graph" is down</p> <p style="padding-left: 40px;"><b>numl</b> entry when "Num Lock" is in effect</p> <p style="padding-left: 40px;"><b>up</b> entry when a key goes up</p> <p>All tables other than <b>up</b> refer to the action generated when a key goes down. Entries in the <b>up</b> table are used only for shift keys, since the shift in question goes away when the key goes up, except for keys such as "Caps Lock" or "Num Lock"; the keyboard streams module makes the key look as if it were a latching key.</p> <p>A table name of <b>all</b> indicates that the entry for all tables should be set to the specified value, with the following exception: for entries with a value other than <b>hole</b>, the entry for the <b>numl</b> table should be set to <b>nonl</b>, and the entry for the <b>up</b> table should be set to <b>nop</b>.</p>

The *code* specifies the effect of the key in question when the specified shift key is down. A *code* consists of either:

- A character, which indicates that the key should generate the given character. The character can either be a single character, a single character preceded by ^ which refers to a "control character" (for instance, ^c is control-C), or a C-style character constant enclosed in single quote characters ('), which can be expressed with C-style escape sequences such as \r for RETURN or \000 for the null character. Note that the single character may be any character in an 8-bit character set, such as ISO 8859/1.
- A string, consisting of a list of characters enclosed in double quote characters ("). Note that the use of the double quote character means that a *code* of double quote must be enclosed in single quotes.
- One of the following expressions:

**shiftkeys+leftshift**

the key is to be the left-hand "Shift" key

**shiftkeys+rightshift**

the key is to be the right-hand "Shift" key

**shiftkeys+leftctrl**

the key is to be the left-hand "Control" key

**shiftkeys+rightctrl**

the key is to be the right-hand "Control" key

**shiftkeys+alt**

the key is to be the "Alt" shift key

**shiftkeys+altgraph**

the key is to be the "Alt Graph" shift key

**shiftkeys+capslock**

the key is to be the "Caps Lock" key

**shiftkeys+shiftlock**

the key is to be the "Shift Lock" key

**shiftkeys+numlock**

the key is to be the "Num Lock" key

**buckybits+systembit**

the key is to be the "Stop" key in SunView; this is normally the L1 key, or the SETUP key on the VT100 keyboard

**buckybits+metabit**

the key is to be the "meta" key. That is, the "Left" or "Right" key on a Sun-2 or Sun-3 keyboard or the "diamond" key on a Sun-4 keyboard

**compose**

the key is to be the "Compose" key

**ctrlq**

on the "VT100" keyboard, the key is to transmit the control-Q character (this would be the entry for the "Q" key in the **ctrl** table)

**ctrls** on the "VT100" keyboard, the key is to transmit the control-S character (this would be the entry for the "S" key in the **ctrl** table)

**noscroll**  
on the "VT100" keyboard, the key is to be the "No Scroll" key

**string+uparrow**  
the key is to be the "up arrow" key

**string+downarrow**  
the key is to be the "down arrow" key

**string+leftarrow**  
the key is to be the "left arrow" key

**string+rightarrow**  
the key is to be the "right arrow" key

**string+homearrow**  
the key is to be the "home" key

**fa\_acute**  
the key is to be the acute accent "floating accent" key

**fa\_cedilla**  
the key is to be the cedilla "floating accent" key

**fa\_cflex**  
the key is to be the circumflex "floating accent" key

**fa\_grave**  
the key is to be the grave accent "floating accent" key

**fa\_tilde**  
the key is to be the tilde "floating accent" key

**fa\_umlaut**  
the key is to be the umlaut "floating accent" key

**nonl** this is used only in the Num Lock table; the key is not to be affected by the state of Num Lock

**pad0** the key is to be the "0" key on the numeric keypad

**pad1** the key is to be the "1" key on the numeric keypad

**pad2** the key is to be the "2" key on the numeric keypad

**pad3** the key is to be the "3" key on the numeric keypad

**pad4** the key is to be the "4" key on the numeric keypad

**pad5** the key is to be the "5" key on the numeric keypad

**pad6** the key is to be the "6" key on the numeric keypad

**pad7** the key is to be the "7" key on the numeric keypad

**pad8** the key is to be the "8" key on the numeric keypad

**pad9** the key is to be the "9" key on the numeric keypad



<b>paddot</b>	the key is to be the "." key on the numeric keypad
<b>padenter</b>	the key is to be the "Enter" key on the numeric keypad
<b>padplus</b>	the key is to be the "+" key on the numeric keypad
<b>padminus</b>	the key is to be the "-" key on the numeric keypad
<b>padstar</b>	the key is to be the "*" key on the numeric keypad
<b>padslash</b>	the key is to be the "/" key on the numeric keypad
<b>padequal</b>	the key is to be the "=" key on the numeric keypad
<b>padsep</b>	the key is to be the "," (separator) key on the numeric keypad
<b>lf(<i>n</i>)</b>	the key is to be the left-hand function key <i>n</i>
<b>rf(<i>n</i>)</b>	the key is to be the right-hand function key <i>n</i>
<b>tf(<i>n</i>)</b>	the key is to be the top function key <i>n</i>
<b>bf(<i>n</i>)</b>	the key is to be the "bottom" function key <i>n</i>
<b>nop</b>	the key is to do nothing
<b>error</b>	this code indicates an internal error; to be used only for keystation 126, and must be used there
<b>idle</b>	this code indicates that the keyboard is idle (that is, has no keys down); to be used only for all entries other than the <b>num1</b> and <b>up</b> table entries for keystation 127, and must be used there
<b>oops</b>	this key exists, but its action is not defined; it has the same effect as <b>nop</b>
<b>reset</b>	this code indicates that the keyboard has just been reset; to be used only for the <b>up</b> table entry for keystation 127, and must be used there.
<b>swap <i>number1</i> with <i>number2</i></b>	exchanges the entries for keystations <i>number1</i> and <i>number2</i> .
<b>key <i>number1</i> same as <i>number2</i></b>	sets the entries for keystation <i>number1</i> to be the same as those for keystation <i>number2</i> . If the file does not specify entries for keystation <i>number2</i> , the entries currently in the translation table are used; if the file does specify entries for keystation <i>number2</i> , those entries are used.

**EXAMPLES**

The following entry sets keystation 15 to be a "hole" (that is, an entry indicating that there is no keystation 15); sets keystation 30 to do nothing when Alt Graph is down, generate "!" when Shift is down, and generate "1" under all other circumstances; and sets keystation 76 to be the left-hand Control key.

```

key 15  all hole
key 30  base 1 shift ! caps 1 ctrl 1 altg nop
key 76  all shiftkeys+leftctrl up shiftkeys+leftctrl

```

The following entry exchanges the Delete and Back Space keys on the Type 4 keyboard:

```
swap 43 with 66
```

Keystation 43 is normally the Back Space key, and keystation 66 is normally the Delete key.

The following entry disables the Caps Lock key on the Type 3 and U.S. Type 4 keyboards:

```
key 119 all nop
```

The following specifies the standard translation tables for the U.S. Type 4 keyboard:

```

key 0  all hole
key 1  all buckybits+systembit up buckybits+systembit
key 2  all hole
key 3  all lf(2)
key 4  all hole
key 5  all tf(1)
key 6  all tf(2)
key 7  all tf(10)
key 8  all tf(3)
key 9  all tf(11)
key 10 all tf(4)
key 11 all tf(12)
key 12 all tf(5)
key 13 all shiftkeys+altgraph up shiftkeys+altgraph
key 14 all tf(6)
key 15 all hole
key 16 all tf(7)
key 17 all tf(8)
key 18 all tf(9)
key 19 all shiftkeys+alt up shiftkeys+alt
key 20 all hole
key 21 all rf(1)
key 22 all rf(2)
key 23 all rf(3)
key 24 all hole
key 25 all lf(3)
key 26 all lf(4)
key 27 all hole
key 28 all hole
key 29 all `[
key 30 base 1 shift ! caps 1 ctrl 1 altg nop
key 31 base 2 shift @ caps 2 ctrl `@ altg nop
key 32 base 3 shift # caps 3 ctrl 3 altg nop

```

```

key 33 base 4 shift $ caps 4 ctrl 4 altg nop
key 34 base 5 shift % caps 5 ctrl 5 altg nop
key 35 base 6 shift ^ caps 6 ctrl ^^ altg nop
key 36 base 7 shift & caps 7 ctrl 7 altg nop
key 37 base 8 shift * caps 8 ctrl 8 altg nop
key 38 base 9 shift ( caps 9 ctrl 9 altg nop
key 39 base 0 shift ) caps 0 ctrl 0 altg nop
key 40 base - shift _ caps - ctrl ^_ altg nop
key 41 base = shift + caps = ctrl = altg nop
key 42 base ' shift ~ caps ' ctrl ^^ altg nop
key 43 all '\b'
key 44 all hole
key 45 all rf(4) numl padequal
key 46 all rf(5) numl padslash
key 47 all rf(6) numl padstar
key 48 all bf(13)
key 49 all lf(5)
key 50 all bf(10) numl padequal
key 51 all lf(6)
key 52 all hole
key 53 all '\t'
key 54 base q shift Q caps Q ctrl ^Q altg nop
key 55 base w shift W caps W ctrl ^W altg nop
key 56 base e shift E caps E ctrl ^E altg nop
key 57 base r shift R caps R ctrl ^R altg nop
key 58 base t shift T caps T ctrl ^T altg nop
key 59 base y shift Y caps Y ctrl ^Y altg nop
key 60 base u shift U caps U ctrl ^U altg nop
key 61 base i shift I caps I ctrl '\t' altg nop
key 62 base o shift O caps O ctrl ^O altg nop
key 63 base p shift P caps P ctrl ^P altg nop
key 64 base [ shift { caps [ ctrl ^[ altg nop
key 65 base ] shift } caps ] ctrl ^] altg nop
key 66 all '\177'
key 67 all compose
key 68 all rf(7) numl pad7
key 69 all rf(8) numl pad8
key 70 all rf(9) numl pad9
key 71 all bf(15) numl padminus
key 72 all lf(7)
key 73 all lf(8)
key 74 all hole
key 75 all hole
key 76 all shiftkeys+leftctrl up shiftkeys+leftctrl
key 77 base a shift A caps A ctrl ^A altg nop

```

```

key 78 base s shift S caps S ctrl ^S altg nop
key 79 base d shift D caps D ctrl ^D altg nop
key 80 base f shift F caps F ctrl ^F altg nop
key 81 base g shift G caps G ctrl ^G altg nop
key 82 base h shift H caps H ctrl ^\b' altg nop
key 83 base j shift J caps J ctrl ^\n' altg nop
key 84 base k shift K caps K ctrl ^\v' altg nop
key 85 base l shift L caps L ctrl ^L altg nop
key 86 base ; shift : caps ; ctrl ; altg nop
key 87 base ` shift ~ caps ` ctrl ^\` altg nop
key 88 base \ shift | caps \ ctrl ^\ altg nop
key 89 all `r'
key 90 all bf(11) numl padenter
key 91 all rf(10) numl pad4
key 92 all rf(11) numl pad5
key 93 all rf(12) numl pad6
key 94 all bf(8) numl pad0
key 95 all lf(9)
key 96 all hole
key 97 all lf(10)
key 98 all shiftkeys+numlock
key 99 all shiftkeys+leftshift up shiftkeys+leftshift
key 100 base z shift Z caps Z ctrl ^Z altg nop
key 101 base x shift X caps X ctrl ^X altg nop
key 102 base c shift C caps C ctrl ^C altg nop
key 103 base v shift V caps V ctrl ^V altg nop
key 104 base b shift B caps B ctrl ^B altg nop
key 105 base n shift N caps N ctrl ^N altg nop
key 106 base m shift M caps M ctrl ^r' altg nop
key 107 base , shift < caps , ctrl , altg nop
key 108 base . shift > caps . ctrl . altg nop
key 109 base / shift ? caps / ctrl ^_ altg nop
key 110 all shiftkeys+rightshift up shiftkeys+rightshift
key 111 all ^n'
key 112 all rf(13) numl pad1
key 113 all rf(14) numl pad2
key 114 all rf(15) numl pad3
key 115 all hole
key 116 all hole
key 117 all hole
key 118 all lf(16)
key 119 all shiftkeys+capslock
key 120 all buckybits+metabit up buckybits+metabit
key 121 base ' shift ' caps ' ctrl ^@ altg '

```

key 122 all buckybits+metabit up buckybits+metabit  
key 123 all hole  
key 124 all hole  
key 125 all bf(14) numl padplus  
key 126 all error numl error up hole  
key 127 all idle numl idle up reset

**ATTRIBUTES**

See **attributes(5)** for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Architecture	SPARC

**SEE ALSO**

**loadkeys(1)**, **attributes(5)**

<b>NAME</b>	krb.conf – Kerberos configuration file
<b>SYNOPSIS</b>	<b>/etc/krb.conf</b>
<b>DESCRIPTION</b>	<p><b>krb.conf</b> contains configuration information describing the Kerberos realm and the Kerberos key distribution center (KDC) servers for known realms.</p> <p><b>krb.conf</b> contains the name of the local realm in the first line, followed by lines indicating realm/host entries. The first token is a realm name, and the second is the hostname of a host running a KDC for that realm. There can be multiple lines for a given realm; the servers are tried in order until an active one is found. The words <i>admin server</i> following the hostname indicate that the host also provides an administrative database server. For example:</p> <pre style="margin-left: 40px;"> ATHENA.MIT.EDU ATHENA.MIT.EDU kerberos-1.mit.edu admin server ATHENA.MIT.EDU kerberos-2.mit.edu LCS.MIT.EDU kerberos.lcs.mit.edu admin server </pre> <p>The Kerberos configuration information can also be supplied using the <b>krb.conf</b> NIS map. If <b>/etc/krb.conf</b> is not found (or the requested information is not found in it), and the system is running NIS, then the information will be obtained from the NIS map. If neither the file nor the NIS map are found, then the Kerberos library will use the <i>domain-name</i> (as returned by <b>domainname(1M)</b>) as the Kerberos realm, and the host <b>kerberos</b> as the location of the KDC. There is no default for the admin server.</p> <p>Note that every time <b>krb.conf</b> is modified, <b>kerbd(1M)</b> needs to be restarted.</p>
<b>SEE ALSO</b>	<b>domainname(1M)</b> , <b>kerbd(1M)</b> , <b>ypmake(1M)</b> , <b>krb.realms(4)</b>
<b>BUGS</b>	There is no NIS+ support yet for the <b>krb.conf</b> map.

<b>NAME</b>	krb.realms – host to Kerberos realm translation file
<b>SYNOPSIS</b>	<code>/etc/krb.realms</code>
<b>DESCRIPTION</b>	<p><b>krb.realms</b> provides a translation from a hostname to the Kerberos realm name for the services provided by that host.</p> <p>Each line of the translation file is in one of the following forms:</p> <p style="padding-left: 40px;"><i>host_name kerberos_realm</i> <i>domain_name kerberos_realm</i></p> <p><i>domain_name</i> should be of the form <i>.XXX.YYY</i>, for example, <b>.LCS.MIT.EDU</b>.</p> <p>If a hostname exactly matches the <i>host_name</i> field in a line of the first form, the corresponding <i>kerberos_realm</i> is used as the realm of the host. If a hostname does not match any <i>host_name</i> in the file, but its domain exactly matches the <i>domain_name</i> field in a line of the second form, the corresponding <i>kerberos_realm</i> is used as the realm of the host.</p> <p>If no translation entry applies, the host's realm is considered to be the hostname's domain portion converted to upper case.</p>
<b>SEE ALSO</b>	<b>krb_realmofhost(3N)</b>
<b>BUGS</b>	There is no NIS or NIS+ support for this information.

<b>NAME</b>	libadm – general administrative library																												
<b>SYNOPSIS</b>	cc [ <i>flag</i> ... ] <i>file</i> ... - <b>ladm</b> [ <i>library</i> ... ]																												
<b>DESCRIPTION</b>	<p>Functions in this library provide Device management, VTOC handling, regular expressions and Packaging routines.</p> <p>The shared object <b>libadm.so.1</b> provides the public interfaces defined below.</p> <p>For additional information on shared object interfaces, see <b>intro(4)</b>.</p>																												
<b>INTERFACES</b>	<p>SUNW_1.1 (generic):</p> <table> <tr> <td><b>advance</b></td> <td><b>asystem</b></td> <td><b>circf</b></td> </tr> <tr> <td><b>compile</b></td> <td><b>devattr</b></td> <td><b>devfree</b></td> </tr> <tr> <td><b>devreserv</b></td> <td><b>getdev</b></td> <td><b>getdgrp</b></td> </tr> <tr> <td><b>getvol</b></td> <td><b>listdev</b></td> <td><b>listdgrp</b></td> </tr> <tr> <td><b>loc1</b></td> <td><b>loc2</b></td> <td><b>locs</b></td> </tr> <tr> <td><b>nbra</b></td> <td><b>pkgdir</b></td> <td><b>pkginfo</b></td> </tr> <tr> <td><b>pkgnmchk</b></td> <td><b>pkgparam</b></td> <td><b>read_vtoc</b></td> </tr> <tr> <td><b>reservdev</b></td> <td><b>sed</b></td> <td><b>step</b></td> </tr> <tr> <td><b>system</b></td> <td><b>write_vtoc</b></td> <td></td> </tr> </table>		<b>advance</b>	<b>asystem</b>	<b>circf</b>	<b>compile</b>	<b>devattr</b>	<b>devfree</b>	<b>devreserv</b>	<b>getdev</b>	<b>getdgrp</b>	<b>getvol</b>	<b>listdev</b>	<b>listdgrp</b>	<b>loc1</b>	<b>loc2</b>	<b>locs</b>	<b>nbra</b>	<b>pkgdir</b>	<b>pkginfo</b>	<b>pkgnmchk</b>	<b>pkgparam</b>	<b>read_vtoc</b>	<b>reservdev</b>	<b>sed</b>	<b>step</b>	<b>system</b>	<b>write_vtoc</b>	
<b>advance</b>	<b>asystem</b>	<b>circf</b>																											
<b>compile</b>	<b>devattr</b>	<b>devfree</b>																											
<b>devreserv</b>	<b>getdev</b>	<b>getdgrp</b>																											
<b>getvol</b>	<b>listdev</b>	<b>listdgrp</b>																											
<b>loc1</b>	<b>loc2</b>	<b>locs</b>																											
<b>nbra</b>	<b>pkgdir</b>	<b>pkginfo</b>																											
<b>pkgnmchk</b>	<b>pkgparam</b>	<b>read_vtoc</b>																											
<b>reservdev</b>	<b>sed</b>	<b>step</b>																											
<b>system</b>	<b>write_vtoc</b>																												
<b>FILES</b>	<p><b>/usr/lib/libadm.so.1</b> shared object</p> <p><b>/usr/lib/libadm.a</b> archive library</p>																												
<b>ATTRIBUTES</b>	See <b>attributes(5)</b> for descriptions of the following attributes:																												
	<table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>MT-Level</td> <td>Unsafe</td> </tr> </tbody> </table>		ATTRIBUTE TYPE	ATTRIBUTE VALUE	MT-Level	Unsafe																							
ATTRIBUTE TYPE	ATTRIBUTE VALUE																												
MT-Level	Unsafe																												
<b>SEE ALSO</b>	<b>pvs(1)</b> , <b>read_vtoc(3X)</b> , <b>system(3)</b> , <b>intro(4)</b> , <b>attributes(5)</b> , <b>regexp(5)</b>																												



<b>NAME</b>	libaio – the asynchronous I/O library																		
<b>SYNOPSIS</b>	cc [ <i>flag</i> ... ] <i>file</i> ... - <b>laio</b> [ <i>library</i> ... ]																		
<b>DESCRIPTION</b>	<p>Functions in this library provide routines for asynchronous I/O. The shared object <b>libaio.so.1</b> provides the public interfaces defined below. For additional information on shared object interfaces, see <b>intro(4)</b>.</p>																		
<b>INTERFACES</b>	<p><b>SISCD_2.3</b> (SPARC only) - The SPARC Compliance Definition, revision 2.3:</p> <table> <tr> <td><b>aiocancel</b></td> <td><b>aioread</b></td> <td><b>aiowait</b></td> </tr> <tr> <td><b>aiowrite</b></td> <td></td> <td></td> </tr> </table> <p><b>SUNW_1.1</b> (generic):</p> <table> <tr> <td><b>aio_close</b></td> <td><b>aio_fork</b></td> <td><b>aioread64</b></td> </tr> <tr> <td><b>aiowrite64</b></td> <td><b>assfail</b></td> <td><b>close</b></td> </tr> <tr> <td><b>fork</b></td> <td><b>sigaction</b></td> <td><b>sigignore</b></td> </tr> <tr> <td><b>signal</b></td> <td><b>sigset</b></td> <td></td> </tr> </table> <p><b>SUNW_1.1</b> (SPARC) - This interface inherits all definitions from the generic <b>SUNW_1.1</b> and the <b>SISCD_2.3</b>.</p> <p><b>SUNW_1.1</b> (i386) - This interface contains all definitions from <b>SISCD_2.3</b>, and inherits all definitions from the generic <b>SUNW_1.1</b>.</p>	<b>aiocancel</b>	<b>aioread</b>	<b>aiowait</b>	<b>aiowrite</b>			<b>aio_close</b>	<b>aio_fork</b>	<b>aioread64</b>	<b>aiowrite64</b>	<b>assfail</b>	<b>close</b>	<b>fork</b>	<b>sigaction</b>	<b>sigignore</b>	<b>signal</b>	<b>sigset</b>	
<b>aiocancel</b>	<b>aioread</b>	<b>aiowait</b>																	
<b>aiowrite</b>																			
<b>aio_close</b>	<b>aio_fork</b>	<b>aioread64</b>																	
<b>aiowrite64</b>	<b>assfail</b>	<b>close</b>																	
<b>fork</b>	<b>sigaction</b>	<b>sigignore</b>																	
<b>signal</b>	<b>sigset</b>																		
<b>FILES</b>	<b>/usr/lib/libaio.so.1</b> shared object																		
<b>ATTRIBUTES</b>	See <b>attributes(5)</b> for descriptions of the following attributes:																		
	<table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>MT-Level</td> <td>Safe</td> </tr> </tbody> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	MT-Level	Safe														
ATTRIBUTE TYPE	ATTRIBUTE VALUE																		
MT-Level	Safe																		
<b>SEE ALSO</b>	<b>pvs(1)</b> , <b>intro(2)</b> , <b>intro(3)</b> , <b>aiocancel(3)</b> , <b>aioread(3)</b> , <b>aiowait(3)</b> , <b>aiowrite(3)</b> , <b>intro(4)</b> , <b>attributes(5)</b>																		

**NAME** libbsm – basic security library

**SYNOPSIS** `cc [ flag ... ] file ... -lbsm [ library ... ]`

**DESCRIPTION** Functions in this library provide basic security, library object reuse and auditing. The shared object **libbsm.so.1** provides the public interfaces defined below. For additional information on shared object interfaces, see **intro(4)**.

**INTERFACES** SUNW\_1.1 (generic):

<code>au_close</code>	<code>audit</code>	<code>auditon</code>
<code>auditsvc</code>	<code>au_open</code>	<code>au_preselect</code>
<code>au_to_arg</code>	<code>au_to_attr</code>	<code>au_to_cmd</code>
<code>au_to_data</code>	<code>au_to_groups</code>	<code>au_to_in_addr</code>
<code>au_to_ipc</code>	<code>au_to_iport</code>	<code>au_to_me</code>
<code>au_to_newgroups</code>	<code>au_to_opaque</code>	<code>au_to_path</code>
<code>au_to_process</code>	<code>au_to_return</code>	<code>au_to_socket</code>
<code>au_to_subject</code>	<code>au_to_text</code>	<code>au_user_mask</code>
<code>au_write</code>	<code>endac</code>	<code>endauclass</code>
<code>endauevent</code>	<code>endauuser</code>	<code>getacdir</code>
<code>getacflg</code>	<code>getacmin</code>	<code>getacna</code>
<code>getauclassent</code>	<code>getauclassent_r</code>	<code>getauclassnam</code>
<code>getauclassnam_r</code>	<code>getaudit</code>	<code>getauditflagsbin</code>
<code>getauditflagschar</code>	<code>getauevent</code>	<code>getauevent_r</code>
<code>getauevnam</code>	<code>getauevnam_r</code>	<code>getauevnonam</code>
<code>getauevnum</code>	<code>getauevnum_r</code>	<code>getaudit</code>
<code>getauuserent</code>	<code>getauuserent_r</code>	<code>getauusernam</code>
<code>getauusernam_r</code>	<code>getfauditflags</code>	<code>setac</code>
<code>setauclass</code>	<code>setauclassfile</code>	<code>setaudit</code>
<code>setauevent</code>	<code>setaueventfile</code>	<code>setaudit</code>
<code>setauuser</code>	<code>setauuserfile</code>	<code>testac</code>

**FILES** `/usr/lib/libbsm.so.1` shared object  
`/usr/lib/libbsm.a` archive library

**ATTRIBUTES** See **attributes(5)** for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT-Level	See individual man page for each function.

**SEE ALSO** **pvs(1)**, **intro(4)**, **attributes(5)**

<b>NAME</b>	libc – the C library																																																																																														
<b>SYNOPSIS</b>	cc [ <i>flag</i> ... ] <i>file</i> ... -lc																																																																																														
<b>DESCRIPTION</b>	<p>Functions in this library provide various facilities defined by System V, ANSI C, POSIX, and so on. See <b>standards(5)</b>. In addition, those facilities previously defined in the internationalization and the wide character libraries are now defined in this library.</p> <p>The shared object <b>libc.so.1</b> provides the public interfaces defined below.</p> <p>For additional information on shared object interfaces, see <b>intro(4)</b>. Many features in this library are implemented upon dynamic linking. Some of these features are not implemented in the archive version.</p>																																																																																														
<b>INTERFACES</b>	<p><b>SYSVABI_1.3</b> (generic) - The System V Application Binary Interface, Third Edition:</p> <table border="0"> <tr> <td><b>abort</b></td> <td><b>abs</b></td> <td><b>_access</b></td> </tr> <tr> <td><b>access</b></td> <td><b>_acct</b></td> <td><b>acct</b></td> </tr> <tr> <td><b>_alarm</b></td> <td><b>alarm</b></td> <td><b>_altzone</b></td> </tr> <tr> <td><b>asctime</b></td> <td><b>__assert</b></td> <td><b>atexit</b></td> </tr> <tr> <td><b>atof</b></td> <td><b>atoi</b></td> <td><b>atol</b></td> </tr> <tr> <td><b>bsearch</b></td> <td><b>calloc</b></td> <td><b>_catclose</b></td> </tr> <tr> <td><b>catclose</b></td> <td><b>_catgets</b></td> <td><b>catgets</b></td> </tr> <tr> <td><b>_catopen</b></td> <td><b>catopen</b></td> <td><b>_cfgetispeed</b></td> </tr> <tr> <td><b>cfgetispeed</b></td> <td><b>_cfgetospeed</b></td> <td><b>cfgetospeed</b></td> </tr> <tr> <td><b>_cfsetispeed</b></td> <td><b>cfsetispeed</b></td> <td><b>_cfsetospeed</b></td> </tr> <tr> <td><b>cfsetospeed</b></td> <td><b>_chdir</b></td> <td><b>chdir</b></td> </tr> <tr> <td><b>_chmod</b></td> <td><b>chmod</b></td> <td><b>_chown</b></td> </tr> <tr> <td><b>chown</b></td> <td><b>_chroot</b></td> <td><b>chroot</b></td> </tr> <tr> <td><b>_cleanup</b></td> <td><b>clearerr</b></td> <td><b>clock</b></td> </tr> <tr> <td><b>_close</b></td> <td><b>close</b></td> <td><b>_closedir</b></td> </tr> <tr> <td><b>closedir</b></td> <td><b>_creat</b></td> <td><b>creat</b></td> </tr> <tr> <td><b>_ctermid</b></td> <td><b>ctermid</b></td> <td><b>ctime</b></td> </tr> <tr> <td><b>__ctype</b></td> <td><b>_cuserid</b></td> <td><b>cuserid</b></td> </tr> <tr> <td><b>_daylight</b></td> <td><b>daylight</b></td> <td><b>difftime</b></td> </tr> <tr> <td><b>div</b></td> <td><b>_dup</b></td> <td><b>dup</b></td> </tr> <tr> <td><b>_dup2</b></td> <td><b>dup2</b></td> <td><b>_environ</b></td> </tr> <tr> <td><b>environ</b></td> <td><b>_execl</b></td> <td><b>execl</b></td> </tr> <tr> <td><b>_execl</b></td> <td><b>execl</b></td> <td><b>_execlp</b></td> </tr> <tr> <td><b>execlp</b></td> <td><b>_execv</b></td> <td><b>execv</b></td> </tr> <tr> <td><b>_execve</b></td> <td><b>execve</b></td> <td><b>_execvp</b></td> </tr> <tr> <td><b>execvp</b></td> <td><b>_exit</b></td> <td><b>exit</b></td> </tr> <tr> <td><b>_fattach</b></td> <td><b>fattach</b></td> <td><b>_fchdir</b></td> </tr> <tr> <td><b>fchdir</b></td> <td><b>_fchmod</b></td> <td><b>fchmod</b></td> </tr> <tr> <td><b>_fchown</b></td> <td><b>fchown</b></td> <td><b>fclose</b></td> </tr> <tr> <td><b>_fcntl</b></td> <td><b>fcntl</b></td> <td><b>_fdetach</b></td> </tr> <tr> <td><b>fdetach</b></td> <td><b>_fdopen</b></td> <td><b>fdopen</b></td> </tr> </table>		<b>abort</b>	<b>abs</b>	<b>_access</b>	<b>access</b>	<b>_acct</b>	<b>acct</b>	<b>_alarm</b>	<b>alarm</b>	<b>_altzone</b>	<b>asctime</b>	<b>__assert</b>	<b>atexit</b>	<b>atof</b>	<b>atoi</b>	<b>atol</b>	<b>bsearch</b>	<b>calloc</b>	<b>_catclose</b>	<b>catclose</b>	<b>_catgets</b>	<b>catgets</b>	<b>_catopen</b>	<b>catopen</b>	<b>_cfgetispeed</b>	<b>cfgetispeed</b>	<b>_cfgetospeed</b>	<b>cfgetospeed</b>	<b>_cfsetispeed</b>	<b>cfsetispeed</b>	<b>_cfsetospeed</b>	<b>cfsetospeed</b>	<b>_chdir</b>	<b>chdir</b>	<b>_chmod</b>	<b>chmod</b>	<b>_chown</b>	<b>chown</b>	<b>_chroot</b>	<b>chroot</b>	<b>_cleanup</b>	<b>clearerr</b>	<b>clock</b>	<b>_close</b>	<b>close</b>	<b>_closedir</b>	<b>closedir</b>	<b>_creat</b>	<b>creat</b>	<b>_ctermid</b>	<b>ctermid</b>	<b>ctime</b>	<b>__ctype</b>	<b>_cuserid</b>	<b>cuserid</b>	<b>_daylight</b>	<b>daylight</b>	<b>difftime</b>	<b>div</b>	<b>_dup</b>	<b>dup</b>	<b>_dup2</b>	<b>dup2</b>	<b>_environ</b>	<b>environ</b>	<b>_execl</b>	<b>execl</b>	<b>_execl</b>	<b>execl</b>	<b>_execlp</b>	<b>execlp</b>	<b>_execv</b>	<b>execv</b>	<b>_execve</b>	<b>execve</b>	<b>_execvp</b>	<b>execvp</b>	<b>_exit</b>	<b>exit</b>	<b>_fattach</b>	<b>fattach</b>	<b>_fchdir</b>	<b>fchdir</b>	<b>_fchmod</b>	<b>fchmod</b>	<b>_fchown</b>	<b>fchown</b>	<b>fclose</b>	<b>_fcntl</b>	<b>fcntl</b>	<b>_fdetach</b>	<b>fdetach</b>	<b>_fdopen</b>	<b>fdopen</b>
<b>abort</b>	<b>abs</b>	<b>_access</b>																																																																																													
<b>access</b>	<b>_acct</b>	<b>acct</b>																																																																																													
<b>_alarm</b>	<b>alarm</b>	<b>_altzone</b>																																																																																													
<b>asctime</b>	<b>__assert</b>	<b>atexit</b>																																																																																													
<b>atof</b>	<b>atoi</b>	<b>atol</b>																																																																																													
<b>bsearch</b>	<b>calloc</b>	<b>_catclose</b>																																																																																													
<b>catclose</b>	<b>_catgets</b>	<b>catgets</b>																																																																																													
<b>_catopen</b>	<b>catopen</b>	<b>_cfgetispeed</b>																																																																																													
<b>cfgetispeed</b>	<b>_cfgetospeed</b>	<b>cfgetospeed</b>																																																																																													
<b>_cfsetispeed</b>	<b>cfsetispeed</b>	<b>_cfsetospeed</b>																																																																																													
<b>cfsetospeed</b>	<b>_chdir</b>	<b>chdir</b>																																																																																													
<b>_chmod</b>	<b>chmod</b>	<b>_chown</b>																																																																																													
<b>chown</b>	<b>_chroot</b>	<b>chroot</b>																																																																																													
<b>_cleanup</b>	<b>clearerr</b>	<b>clock</b>																																																																																													
<b>_close</b>	<b>close</b>	<b>_closedir</b>																																																																																													
<b>closedir</b>	<b>_creat</b>	<b>creat</b>																																																																																													
<b>_ctermid</b>	<b>ctermid</b>	<b>ctime</b>																																																																																													
<b>__ctype</b>	<b>_cuserid</b>	<b>cuserid</b>																																																																																													
<b>_daylight</b>	<b>daylight</b>	<b>difftime</b>																																																																																													
<b>div</b>	<b>_dup</b>	<b>dup</b>																																																																																													
<b>_dup2</b>	<b>dup2</b>	<b>_environ</b>																																																																																													
<b>environ</b>	<b>_execl</b>	<b>execl</b>																																																																																													
<b>_execl</b>	<b>execl</b>	<b>_execlp</b>																																																																																													
<b>execlp</b>	<b>_execv</b>	<b>execv</b>																																																																																													
<b>_execve</b>	<b>execve</b>	<b>_execvp</b>																																																																																													
<b>execvp</b>	<b>_exit</b>	<b>exit</b>																																																																																													
<b>_fattach</b>	<b>fattach</b>	<b>_fchdir</b>																																																																																													
<b>fchdir</b>	<b>_fchmod</b>	<b>fchmod</b>																																																																																													
<b>_fchown</b>	<b>fchown</b>	<b>fclose</b>																																																																																													
<b>_fcntl</b>	<b>fcntl</b>	<b>_fdetach</b>																																																																																													
<b>fdetach</b>	<b>_fdopen</b>	<b>fdopen</b>																																																																																													

<b>feof</b>	<b>ferror</b>	<b>fflush</b>
<b>fgetc</b>	<b>fgetpos</b>	<b>fgets</b>
<b>__filbuf</b>	<b>_fileno</b>	<b>fileno</b>
<b>__flsbuf</b>	<b>_fmtmsg</b>	<b>fmtmsg</b>
<b>fopen</b>	<b>_fork</b>	<b>fork</b>
<b>_fpathconf</b>	<b>fpathconf</b>	<b>fprintf</b>
<b>fputc</b>	<b>fputs</b>	<b>fread</b>
<b>free</b>	<b>freopen</b>	<b>frexp</b>
<b>fscanf</b>	<b>fseek</b>	<b>fsetpos</b>
<b>_fstat</b>	<b>fstat</b>	<b>_fstatvfs</b>
<b>fstatvfs</b>	<b>_fsync</b>	<b>fsync</b>
<b>ftell</b>	<b>_ftok</b>	<b>ftok</b>
<b>fwrite</b>	<b>getc</b>	<b>getchar</b>
<b>_getcontext</b>	<b>getcontext</b>	<b>_getcwd</b>
<b>getcwd</b>	<b>_getdate</b>	<b>getdate</b>
<b>_getdate_err</b>	<b>getdate_err</b>	<b>_getegid</b>
<b>getegid</b>	<b>getenv</b>	<b>_geteuid</b>
<b>geteuid</b>	<b>_getgid</b>	<b>getgid</b>
<b>_getgrgid</b>	<b>getgrgid</b>	<b>_getgrnam</b>
<b>getgrnam</b>	<b>_getgroups</b>	<b>getgroups</b>
<b>_getlogin</b>	<b>getlogin</b>	<b>_getmsg</b>
<b>getmsg</b>	<b>_getopt</b>	<b>getopt</b>
<b>_getpass</b>	<b>getpass</b>	<b>_getpgid</b>
<b>getpgid</b>	<b>_getpgrp</b>	<b>getpgrp</b>
<b>_getpid</b>	<b>getpid</b>	<b>_getpmsg</b>
<b>getpmsg</b>	<b>_getppid</b>	<b>getppid</b>
<b>_getpwnam</b>	<b>getpwnam</b>	<b>_getpwuid</b>
<b>getpwuid</b>	<b>_getrlimit</b>	<b>getrlimit</b>
<b>gets</b>	<b>_getsid</b>	<b>getsid</b>
<b>_getsubopt</b>	<b>getsubopt</b>	<b>_gettxt</b>
<b>gettxt</b>	<b>_getuid</b>	<b>getuid</b>
<b>_getw</b>	<b>getw</b>	<b>gmtime</b>
<b>_grantpt</b>	<b>grantpt</b>	<b>_hcreate</b>
<b>hcreate</b>	<b>_hdestroy</b>	<b>hdestroy</b>
<b>_hsearch</b>	<b>hsearch</b>	<b>_initgroups</b>
<b>initgroups</b>	<b>_iob</b>	<b>_ioctl</b>
<b>ioctl</b>	<b>isalnum</b>	<b>isalpha</b>
<b>_isascii</b>	<b>isascii</b>	<b>_isastream</b>
<b>isastream</b>	<b>_isatty</b>	<b>isatty</b>
<b>iscntrl</b>	<b>isdigit</b>	<b>isgraph</b>
<b>islower</b>	<b>_isnan</b>	<b>isnan</b>
<b>_isnand</b>	<b>isnand</b>	<b>isprint</b>
<b>ispunct</b>	<b>isspace</b>	<b>isupper</b>
<b>isxdigit</b>	<b>_kill</b>	<b>kill</b>
<b>labs</b>	<b>_lchown</b>	<b>lchown</b>

<b>ldexp</b>	<b>ldiv</b>	<b>_lfind</b>
<b>lfind</b>	<b>_link</b>	<b>link</b>
<b>localeconv</b>	<b>localtime</b>	<b>_lockf</b>
<b>lockf</b>	<b>logb</b>	<b>longjmp</b>
<b>_lsearch</b>	<b>lsearch</b>	<b>_lseek</b>
<b>lseek</b>	<b>_lstat</b>	<b>lstat</b>
<b>_makecontext</b>	<b>makecontext</b>	<b>malloc</b>
<b>mblen</b>	<b>mbstowcs</b>	<b>mbtowc</b>
<b>_memccpy</b>	<b>memccpy</b>	<b>memchr</b>
<b>memcmp</b>	<b>_memcntl</b>	<b>memcntl</b>
<b>memcpy</b>	<b>memmove</b>	<b>memset</b>
<b>_mkdir</b>	<b>mkdir</b>	<b>_mkfifo</b>
<b>mkfifo</b>	<b>_mknod</b>	<b>mknod</b>
<b>_mktemp</b>	<b>mktemp</b>	<b>mktime</b>
<b>_mlock</b>	<b>mlock</b>	<b>_mmap</b>
<b>mmap</b>	<b>_modf</b>	<b>modf</b>
<b>_monitor</b>	<b>monitor</b>	<b>_mount</b>
<b>mount</b>	<b>_mprotect</b>	<b>mprotect</b>
<b>_msgctl</b>	<b>msgctl</b>	<b>_msgget</b>
<b>msgget</b>	<b>_msgrcv</b>	<b>msgrcv</b>
<b>_msgsnd</b>	<b>msgsnd</b>	<b>_msync</b>
<b>msync</b>	<b>_munlock</b>	<b>munlock</b>
<b>_munmap</b>	<b>munmap</b>	<b>_nextafter</b>
<b>nextafter</b>	<b>_nftw</b>	<b>nftw</b>
<b>_nice</b>	<b>nice</b>	<b>_nl_langinfo</b>
<b>nl_langinfo</b>	<b>_numeric</b>	<b>_open</b>
<b>open</b>	<b>_opendir</b>	<b>opendir</b>
<b>optarg</b>	<b>opterr</b>	<b>optind</b>
<b>optopt</b>	<b>_pathconf</b>	<b>pathconf</b>
<b>_pause</b>	<b>pause</b>	<b>_pclose</b>
<b>pclose</b>	<b>perror</b>	<b>_pipe</b>
<b>pipe</b>	<b>_poll</b>	<b>poll</b>
<b>_popen</b>	<b>popen</b>	<b>printf</b>
<b>_profil</b>	<b>profil</b>	<b>_ptrace</b>
<b>ptrace</b>	<b>_ptsname</b>	<b>ptsname</b>
<b>putc</b>	<b>putchar</b>	<b>_putenv</b>
<b>putenv</b>	<b>_putmsg</b>	<b>putmsg</b>
<b>_putpmsg</b>	<b>putpmsg</b>	<b>puts</b>
<b>_putw</b>	<b>putw</b>	<b>qsort</b>
<b>raise</b>	<b>rand</b>	<b>_read</b>
<b>read</b>	<b>_readdir</b>	<b>readdir</b>
<b>_readlink</b>	<b>readlink</b>	<b>_readv</b>
<b>readv</b>	<b>realloc</b>	<b>remove</b>
<b>_rename</b>	<b>rename</b>	<b>rewind</b>
<b>_rewinddir</b>	<b>rewinddir</b>	<b>_rmdir</b>

<b>rmdir</b>	<b>_scalb</b>	<b>scalb</b>
<b>scanf</b>	<b>_seekdir</b>	<b>seekdir</b>
<b>_semctl</b>	<b>semctl</b>	<b>_semget</b>
<b>semget</b>	<b>_semop</b>	<b>semop</b>
<b>setbuf</b>	<b>_setcontext</b>	<b>setcontext</b>
<b>_setgid</b>	<b>setgid</b>	<b>_setgroups</b>
<b>setgroups</b>	<b>setjmp</b>	<b>setlabel</b>
<b>setlocale</b>	<b>_setpgid</b>	<b>setpgid</b>
<b>_setpgrp</b>	<b>setpgrp</b>	<b>_setrlimit</b>
<b>setrlimit</b>	<b>_setsid</b>	<b>setsid</b>
<b>_setuid</b>	<b>setuid</b>	<b>setvbuf</b>
<b>_shmat</b>	<b>shmat</b>	<b>_shmctl</b>
<b>shmctl</b>	<b>_shmdt</b>	<b>shmdt</b>
<b>_shmget</b>	<b>shmget</b>	<b>_sigaction</b>
<b>sigaction</b>	<b>_sigaddset</b>	<b>sigaddset</b>
<b>_sigaltstack</b>	<b>sigaltstack</b>	<b>_sigdelset</b>
<b>sigdelset</b>	<b>_sigemptyset</b>	<b>sigemptyset</b>
<b>_sigfillset</b>	<b>sigfillset</b>	<b>_sighold</b>
<b>sighold</b>	<b>_sigignore</b>	<b>sigignore</b>
<b>_sigismember</b>	<b>sigismember</b>	<b>_siglongjmp</b>
<b>siglongjmp</b>	<b>signal</b>	<b>_sigpause</b>
<b>sigpause</b>	<b>_sigpending</b>	<b>sigpending</b>
<b>_sigprocmask</b>	<b>sigprocmask</b>	<b>_sigrelse</b>
<b>sigrelse</b>	<b>_sigsend</b>	<b>sigsend</b>
<b>_sigsendset</b>	<b>sigsendset</b>	<b>_sigset</b>
<b>sigset</b>	<b>_sigsetjmp</b>	<b>sigsetjmp</b>
<b>_sigsuspend</b>	<b>sigsuspend</b>	<b>_sleep</b>
<b>sleep</b>	<b>sprintf</b>	<b>rand</b>
<b>sscanf</b>	<b>_stat</b>	<b>stat</b>
<b>_statvfs</b>	<b>statvfs</b>	<b>_stime</b>
<b>stime</b>	<b>strcat</b>	<b>strchr</b>
<b>strcmp</b>	<b>strcoll</b>	<b>strcpy</b>
<b>strcsn</b>	<b>_strdup</b>	<b>strdup</b>
<b>strerror</b>	<b>strftime</b>	<b>strlen</b>
<b>strncat</b>	<b>strncmp</b>	<b>strncpy</b>
<b>strpbrk</b>	<b>strchr</b>	<b>strspn</b>
<b>strstr</b>	<b>strtod</b>	<b>strtok</b>
<b>strtol</b>	<b>strtoul</b>	<b>strxfrm</b>
<b>_swab</b>	<b>swab</b>	<b>_swapcontext</b>
<b>swapcontext</b>	<b>_symlink</b>	<b>symlink</b>
<b>_sync</b>	<b>sync</b>	<b>_sysconf</b>
<b>sysconf</b>	<b>system</b>	<b>_tcdrain</b>
<b>tcdrain</b>	<b>_tcfow</b>	<b>tcflow</b>
<b>_tcflush</b>	<b>tcflush</b>	<b>_tcgetattr</b>
<b>tcgetattr</b>	<b>_tcgetpgrp</b>	<b>tcgetpgrp</b>

<b>_tcgetsid</b>	<b>tcgetsid</b>	<b>_tcsendbreak</b>
<b>tcsendbreak</b>	<b>_tcsetattr</b>	<b>tcsetattr</b>
<b>_tcsetpgrp</b>	<b>tcsetpgrp</b>	<b>_tdelete</b>
<b>tdelete</b>	<b>_tell</b>	<b>tell</b>
<b>_telldir</b>	<b>telldir</b>	<b>_tempnam</b>
<b>tempnam</b>	<b>_tfind</b>	<b>tfind</b>
<b>_time</b>	<b>time</b>	<b>_times</b>
<b>times</b>	<b>_timezone</b>	<b>timezone</b>
<b>tmpfile</b>	<b>tmpnam</b>	<b>_toascii</b>
<b>toascii</b>	<b>_tolower</b>	<b>tolower</b>
<b>_toupper</b>	<b>toupper</b>	<b>_tsearch</b>
<b>tsearch</b>	<b>_ttyname</b>	<b>ttyname</b>
<b>_twalk</b>	<b>twalk</b>	<b>_tzname</b>
<b>tzname</b>	<b>_tzset</b>	<b>tzset</b>
<b>_ulimit</b>	<b>ulimit</b>	<b>_umask</b>
<b>umask</b>	<b>_umount</b>	<b>umount</b>
<b>_uname</b>	<b>uname</b>	<b>ungetc</b>
<b>_unlink</b>	<b>unlink</b>	<b>_unlockpt</b>
<b>unlockpt</b>	<b>_utime</b>	<b>utime</b>
<b>vsprintf</b>	<b>vprintf</b>	<b>vsprintf</b>
<b>_wait</b>	<b>wait</b>	<b>_waitid</b>
<b>waitid</b>	<b>_waitpid</b>	<b>waitpid</b>
<b>wcstombs</b>	<b>wctomb</b>	<b>_write</b>
<b>write</b>	<b>_writev</b>	<b>writev</b>
<b>_xftw</b>		

**SYSVABI\_1.3 (SPARC) -**

The SPARC Processor Supplement. This interface contains all of the generic SYSVABI\_1.3, and defines:

<b>_Q_add</b>	<b>_Q_cmp</b>	<b>_Q_cmpe</b>
<b>_Q_div</b>	<b>_Q_dtoq</b>	<b>_Q_feq</b>
<b>_Q_fge</b>	<b>_Q_fgt</b>	<b>_Q_fle</b>
<b>_Q_fit</b>	<b>_Q_fne</b>	<b>_Q_itoq</b>
<b>_Q_mul</b>	<b>_Q_neg</b>	<b>_Q_qtod</b>
<b>_Q_qtoi</b>	<b>_Q_qtos</b>	<b>_Q_qtou</b>
<b>_Q_sqrt</b>	<b>_Q_stoq</b>	<b>_Q_sub</b>
<b>_Q_utoq</b>	<b>.div</b>	<b>__dtou</b>
<b>__ftou</b>	<b>__huge_val</b>	<b>.mul</b>
<b>.rem</b>	<b>.stret1</b>	<b>.stret2</b>
<b>.stret4</b>	<b>.stret8</b>	<b>.udiv</b>
<b>.umul</b>	<b>.urem</b>	

**SYSVABI\_1.3 (i386) -**

The Intel386 Processor Supplement. This interface contains all of the generic SYSVABI\_1.3, and defines:

<b>__flt_rounds</b>	<b>__fp_hw</b>	<b>__fpstart</b>
<b>__fpstart</b>	<b>__fxstat</b>	<b>__huge_val</b>
<b>__lxstat</b>	<b>__nuname</b>	<b>nuname</b>
<b>__sbrk</b>	<b>sbrk</b>	<b>__xmknod</b>
<b>__xstat</b>		

**SISCD\_2.3 (SPARC only) -**

The SPARC Compliance Definition, revision 2.3. This interface inherits all definitions from SYSVABI\_1.3, and defines:

<b>__addseverity</b>	<b>addseverity</b>	<b>asctime_r</b>
<b>__crypt</b>	<b>crypt</b>	<b>ctime_r</b>
<b>__div64</b>	<b>__dtoll</b>	<b>__dtoull</b>
<b>__encrypt</b>	<b>encrypt</b>	<b>endgrent</b>
<b>__endpwent</b>	<b>__errno</b>	<b>errno</b>
<b>fgetgrent</b>	<b>fgetgrent_r</b>	<b>fgetpwent</b>
<b>fgetpwent_r</b>	<b>flockfile</b>	<b>__ftoll</b>
<b>__ftoull</b>	<b>funlockfile</b>	<b>getchar_unlocked</b>
<b>getc_unlocked</b>	<b>getgrent</b>	<b>getgrent_r</b>
<b>getgrgid_r</b>	<b>getgrnam_r</b>	<b>__getitimer</b>
<b>getitimer</b>	<b>getlogin_r</b>	<b>getpwent</b>
<b>getpwent_r</b>	<b>getpwnam_r</b>	<b>getpwuid_r</b>
<b>__gettimeofday</b>	<b>gettimeofday</b>	<b>gmtime_r</b>
<b>__job</b>	<b>localtime_r</b>	<b>__mul64</b>
<b>putchar_unlocked</b>	<b>putc_unlocked</b>	<b>rand_r</b>
<b>readdir_r</b>	<b>__rem64</b>	<b>__sbrk</b>
<b>sbrk</b>	<b>setgrent</b>	<b>__setitimer</b>
<b>setitimer</b>	<b>__setkey</b>	<b>setkey</b>
<b>setpwent</b>	<b>strtok_r</b>	<b>__sysinfo</b>
<b>sysinfo</b>	<b>ttynam_r</b>	<b>__udiv64</b>
<b>__umul64</b>	<b>__urem64</b>	

**SUNW\_1.1 (generic):**

<b>a64l</b>	<b>acl</b>
<b>addsev</b>	<b>adjtime</b>
<b>altzone</b>	<b>ascftime</b>
<b>__assert</b>	<b>atoll</b>
<b>bcmp</b>	<b>bcopy</b>
<b>brk</b>	<b>__bufendtab</b>
<b>__builtin_alloca</b>	<b>bzero</b>
<b>cfree</b>	<b>ctime</b>
<b>closelog</b>	<b>cond_broadcast</b>
<b>cond_destroy</b>	<b>cond_init</b>



<code>cond_signal</code>	<code>cond_timedwait</code>
<code>cond_wait</code>	<code>confstr</code>
<code>csetcol</code>	<code>csetlen</code>
<code>ctermid_r</code>	<code>_ctype</code>
<code>dbm_close</code>	<code>dbm_delete</code>
<code>dbm_fetch</code>	<code>dbm_firstkey</code>
<code>dbm_nextkey</code>	<code>dbm_open</code>
<code>dbm_store</code>	<code>decimal_to_double</code>
<code>decimal_to_extended</code>	<code>decimal_to_quadruple</code>
<code>decimal_to_single</code>	<code>double_to_decimal</code>
<code>drand48</code>	<code>econvert</code>
<code>ecvt</code>	<code>endnetgrent</code>
<code>endspent</code>	<code>endusershell</code>
<code>endutent</code>	<code>endutxent</code>
<code>erand48</code>	<code>euccol</code>
<code>euclen</code>	<code>eucscol</code>
<code>_exithandle</code>	<code>exportfs</code>
<code>extended_to_decimal</code>	<code>fac1</code>
<code>fchroot</code>	<code>fconvert</code>
<code>fcvt</code>	<code>ffs</code>
<code>fgetspent</code>	<code>fgetspent_r</code>
<code>_filbuf</code>	<code>file_to_decimal</code>
<code>finite</code>	<code>_flsbuf</code>
<code>fnmatch</code>	<code>fork1</code>
<code>fpclass</code>	<code>fpgetmask</code>
<code>fpgetround</code>	<code>fpgetsticky</code>
<code>fpsetmask</code>	<code>fpsetround</code>
<code>fpsetsticky</code>	<code>fstatfs</code>
<code>ftime</code>	<code>ftruncate</code>
<code>ftw</code>	<code>func_to_decimal</code>
<code>gconvert</code>	<code>gcvt</code>
<code>_getdate_err_addr</code>	<code>getdents</code>
<code>getdtablesize</code>	<code>gethostid</code>
<code>gethostname</code>	<code>gethrtime</code>
<code>gethrvtime</code>	<code>getmntany</code>
<code>getmntent</code>	<code>getnetgrent</code>
<code>getnetgrent_r</code>	<code>getpagesize</code>
<code>getpriority</code>	<code>getpw</code>
<code>getrusage</code>	<code>getspent</code>
<code>getspent_r</code>	<code>getspnam</code>
<code>getspnam_r</code>	<code>getusershell</code>
<code>getutent</code>	<code>getutid</code>
<code>getutline</code>	<code>getutmp</code>
<code>getutmpx</code>	<code>getutxent</code>
<code>getutxid</code>	<code>getutxline</code>

getvfsany	getvfsent
getvfsfile	getvfsspec
getwd	getwidth
glob	globfree
gsignal	hasmntopt
iconv	iconv_close
iconv_open	index
initstate	innetgr
_insque	insque
isnanf	jrand48
killpg	l64a
ladd	_lastbuf
lckpwwdf	lcong48
ldivide	lexp10
lfmt	llabs
lldiv	llog10
llseek	lltostr
lmul	lone
lrnd48	lshiffl
lsub	lten
_lwp_cond_broadcast	_lwp_cond_signal
_lwp_cond_timedwait	_lwp_cond_wait
_lwp_continue	_lwp_create
_lwp_exit	_lwp_getprivate
_lwp_info	_lwp_kill
_lwp_makecontext	_lwp_mutex_lock
_lwp_mutex_trylock	_lwp_mutex_unlock
_lwp_self	_lwp_sema_init
_lwp_sema_post	_lwp_sema_wait
_lwp_setprivate	_lwp_suspend
_lwp_wait	lzero
madvise	__major
__makedev	makeutx
memalign	mincore
__minor	mlockall
modctl	modff
modutx	mrnd48
munlockall	mutex_destroy
_mutex_held	mutex_init
_mutex_lock	mutex_lock
mutex_trylock	mutex_unlock
nfs_getfh	nrnd48
_nsc_trydoorcall	_nss_XbyY_buf_alloc
_nss_XbyY_buf_free	nss_default_finders
nss_delete	nss_endent

<b>nss_getent</b>	<b>_nss_netdb_aliases</b>
<b>nss_search</b>	<b>nss_setent</b>
<b>__nsw_extended_action</b>	<b>__nsw_freeconfig</b>
<b>__nsw_getconfig</b>	<b>openlog</b>
<b>pfmt</b>	<b>plock</b>
<b>p_online</b>	<b>__posix_asctime_r</b>
<b>__posix_ctime_r</b>	<b>__posix_getgrgid_r</b>
<b>__posix_getgrnam_r</b>	<b>__posix_getlogin_r</b>
<b>__posix_getpwnam_r</b>	<b>__posix_getpwuid_r</b>
<b>__posix_readdir_r</b>	<b>__posix_sigwait</b>
<b>__posix_ttyname_r</b>	<b>pread</b>
<b>__priocntl</b>	<b>__priocntlset</b>
<b>processor_bind</b>	<b>processor_info</b>
<b>psiginfo</b>	<b>psignal</b>
<b>pthread_condattr_destroy</b>	<b>pthread_condattr_getpshared</b>
<b>pthread_condattr_init</b>	<b>pthread_condattr_setpshared</b>
<b>pthread_cond_broadcast</b>	<b>pthread_cond_destroy</b>
<b>pthread_cond_init</b>	<b>pthread_cond_signal</b>
<b>pthread_cond_timedwait</b>	<b>pthread_cond_wait</b>
<b>pthread_mutexattr_destroy</b>	<b>pthread_mutexattr_getprioceiling</b>
<b>pthread_mutexattr_getprotocol</b>	<b>pthread_mutexattr_getpshared</b>
<b>pthread_mutexattr_init</b>	<b>pthread_mutexattr_setprioceiling</b>
<b>pthread_mutexattr_setprotocol</b>	<b>pthread_mutexattr_setpshared</b>
<b>pthread_mutex_destroy</b>	<b>pthread_mutex_getprioceiling</b>
<b>pthread_mutex_init</b>	<b>pthread_mutex_lock</b>
<b>pthread_mutex_setprioceiling</b>	<b>pthread_mutex_trylock</b>
<b>pthread_mutex_unlock</b>	<b>putpwent</b>
<b>putspent</b>	<b>pututline</b>
<b>pututxline</b>	<b>pwrite</b>
<b>qeconvert</b>	<b>qecvt</b>
<b>qfconvert</b>	<b>qfcvt</b>
<b>qgconvert</b>	<b>qgcvt</b>
<b>quadruple_to_decimal</b>	<b>random</b>
<b>realpath</b>	<b>reboot</b>
<b>re_comp</b>	<b>re_exec</b>
<b>regcomp</b>	<b>regerror</b>
<b>regexec</b>	<b>regfree</b>
<b>_remque</b>	<b>remque</b>
<b>rindex</b>	<b>rwlock_init</b>
<b>rw_rdlock</b>	<b>_rw_read_held</b>
<b>rw_read_held</b>	<b>rw_tryrdlock</b>
<b>rw_trywrlock</b>	<b>rw_unlock</b>
<b>_rw_write_held</b>	<b>rw_write_held</b>
<b>rw_wrlock</b>	<b>seconvert</b>
<b>seed48</b>	<b>select</b>

<b>_sema_held</b>	<b>sema_held</b>
<b>sema_init</b>	<b>sema_post</b>
<b>sema_trywait</b>	<b>sema_wait</b>
<b>setbuffer</b>	<b>setcat</b>
<b>setegid</b>	<b>seteuid</b>
<b>sethostname</b>	<b>setlinebuf</b>
<b>setlogmask</b>	<b>setnetgrent</b>
<b>setpriority</b>	<b>setregid</b>
<b>setreuid</b>	<b>setspent</b>
<b>setstate</b>	<b>settimeofday</b>
<b>setusershell</b>	<b>setutent</b>
<b>setutxent</b>	<b>sfconvert</b>
<b>sgconvert</b>	<b>_sibuf</b>
<b>sig2str</b>	<b>sigfpe</b>
<b>sigwait</b>	<b>single_to_decimal</b>
<b>_sobuf</b>	<b>rand48</b>
<b>srandom</b>	<b>ssignal</b>
<b>statfs</b>	<b>str2sig</b>
<b>strcasecmp</b>	<b>strfmon</b>
<b>string_to_decimal</b>	<b>strncasecmp</b>
<b>strptime</b>	<b>strsignal</b>
<b>strtoll</b>	<b>strtoull</b>
<b>swapctl</b>	<b>sync_instruction_memory</b>
<b>_sys_buslist</b>	<b>_syscall</b>
<b>syscall</b>	<b>_sys_cldlist</b>
<b>_sys_fpelist</b>	<b>sysfs</b>
<b>_sys_illlist</b>	<b>_syslog</b>
<b>syslog</b>	<b>_sys_nsig</b>
<b>_sys_segvlst</b>	<b>_sys_siginfo</b>
<b>_sys_siglist</b>	<b>_sys_siglistn</b>
<b>_sys_siglistp</b>	<b>_sys_traplist</b>
<b>thr_continue</b>	<b>thr_create</b>
<b>thr_exit</b>	<b>thr_getconcurrency</b>
<b>thr_getprio</b>	<b>thr_getspecific</b>
<b>thr_join</b>	<b>thr_keycreate</b>
<b>thr_kill</b>	<b>thr_min_stack</b>
<b>thr_self</b>	<b>thr_setconcurrency</b>
<b>thr_setprio</b>	<b>thr_setspecific</b>
<b>thr_sigsetmask</b>	<b>thr_stksegment</b>
<b>thr_suspend</b>	<b>thr_yield</b>
<b>tmpnam_r</b>	<b>truncate</b>
<b>ttyslot</b>	<b>uadmin</b>
<b>ualarm</b>	<b>ulckpwn</b>
<b>ulltostr</b>	<b>unordered</b>
<b>updwtmp</b>	<b>updwtmpx</b>

<b>usleep</b>	<b>ustat</b>
<b>utimes</b>	<b>utmpname</b>
<b>utmpxname</b>	<b>valloc</b>
<b>vfork</b>	<b>vhangup</b>
<b>vlfmt</b>	<b>vpfmt</b>
<b>vsyslog</b>	<b>wait3</b>
<b>wait4</b>	<b>wordexp</b>
<b>wordfree</b>	<b>__xpg4</b>
<b>yield</b>	

**SUNW\_1.1 (SPARC) -**

This interface inherits all definitions from the generic SUNW\_1.1 and the SISCD\_2.3, and defines:

**\_\_flt\_rounds**

**SUNW\_1.1 (i386) -**

This interface contains all definitions from SISCD\_2.3, inherits all definitions from the generic SUNW\_1.1 and the SYSVABI\_1.3, and defines:

**\_thr\_errno\_addr**

**SUNW\_1.2 - SUNW\_1.16 (generic) -**

These interfaces inherit all definitions from the generic SUNW\_1.1, and define:

<b>basename</b>	<b>bindtextdomain</b>
<b>bsd_signal</b>	<b>_creat64</b>
<b>creat64</b>	<b>dbm_clearerr</b>
<b>dbm_error</b>	<b>dcgettext</b>
<b>dgettext</b>	<b>directio</b>
<b>dirname</b>	<b>fgetpos64</b>
<b>fgetwc</b>	<b>fgetws</b>
<b>fopen64</b>	<b>fputwc</b>
<b>fputws</b>	<b>freopen64</b>
<b>fseeko</b>	<b>fseeko64</b>
<b>fsetpos64</b>	<b>_fstat64</b>
<b>fstat64</b>	<b>_fstatvfs64</b>
<b>fstatvfs64</b>	<b>ftello</b>
<b>ftello64</b>	<b>_ftruncate64</b>
<b>ftruncate64</b>	<b>_ftw64</b>
<b>ftw64</b>	<b>_getdents64</b>
<b>getdents64</b>	<b>_getexecname</b>
<b>getexecname</b>	<b>getpassphrase</b>
<b>_getrlimit64</b>	<b>getrlimit64</b>
<b>gettext</b>	<b>getwc</b>
<b>getwchar</b>	<b>getws</b>
<b>isenglish</b>	<b>isideogram</b>
<b>isnumber</b>	<b>isphonogram</b>

isspecial	iswalnum
iswalpha	iswcntrl
iswctype	iswdigit
iswgraph	iswlower
iswprint	iswpunct
iswspace	iswupper
iswxdigit	__loc1
_lockf64	lockf64
_longjmp	_lseek64
lseek64	_lstat64
lstat64	_lwp_sema_trywait
_mkstemp64	mkstemp64
_mmap64	mmap64
_nftw64	nftw64
_ntp_adjtime	ntp_adjtime
_ntp_gettime	ntp_gettime
_open64	open64
_pread64	pread64
pset_assign	pset_bind
pset_create	pset_destroy
pset_info	pthread_atfork
pthread_attr_destroy	pthread_attr_getdetachstate
pthread_attr_getinheritsched	pthread_attr_getschedparam
pthread_attr_getschedpolicy	pthread_attr_getscope
pthread_attr_getstackaddr	pthread_attr_getstacksize
pthread_attr_init	pthread_attr_setdetachstate
pthread_attr_setinheritsched	pthread_attr_setschedparam
pthread_attr_setschedpolicy	pthread_attr_setscope
pthread_attr_setstackaddr	pthread_attr_setstacksize
pthread_cancel	__pthread_cleanup_pop
__pthread_cleanup_push	pthread_create
pthread_detach	pthread_equal
pthread_exit	pthread_getschedparam
pthread_getspecific	pthread_join
pthread_key_create	pthread_key_delete
pthread_kill	pthread_once
pthread_self	pthread_setcancelstate
pthread_setcanceltype	pthread_setschedparam
pthread_setspecific	pthread_sigmask
pthread_testcancel	putwc
putwchar	putws
_pwrite64	pwrite64
_readdir64	readdir64
_readdir64_r	readdir64_r
regcmp	regex

<b>_resolvepath</b>	<b>resolvepath</b>
<b>_rwlck_destroy</b>	<b>rwlock_destroy</b>
<b>_sema_destroy</b>	<b>sema_destroy</b>
<b>_setjmp</b>	<b>_setrlimit64</b>
<b>setrlimit64</b>	<b>_s_fcntl</b>
<b>s_fcntl</b>	<b>siginterrupt</b>
<b>sigstack</b>	<b>s_ioctl</b>
<b>snprintf</b>	<b>_stat64</b>
<b>stat64</b>	<b>_statvfs64</b>
<b>statvfs64</b>	<b>strtows</b>
<b>textdomain</b>	<b>tmpfile64</b>
<b>towctrans</b>	<b>towlower</b>
<b>towupper</b>	<b>_truncate64</b>
<b>truncate64</b>	<b>ungetwc</b>
<b>vsprintf</b>	<b>watoll</b>
<b>wscat</b>	<b>wcschr</b>
<b>wscmp</b>	<b>wcscoll</b>
<b>wscopy</b>	<b>wcscspn</b>
<b>wcsftime</b>	<b>wcslen</b>
<b>wcsncat</b>	<b>wcsncmp</b>
<b>wcsncpy</b>	<b>wcspbrk</b>
<b>wcsrchr</b>	<b>wcsspn</b>
<b>wcstod</b>	<b>wcstok</b>
<b>wcstol</b>	<b>wcstoul</b>
<b>wcswcs</b>	<b>wcswidth</b>
<b>wcsxfrm</b>	<b>wctrans</b>
<b>wctype</b>	<b>wcwidth</b>
<b>wscasecmp</b>	<b>wscat</b>
<b>wchr</b>	<b>wscmp</b>
<b>wscol</b>	<b>wscoll</b>
<b>wscopy</b>	<b>wcspn</b>
<b>wsdup</b>	<b>wslen</b>
<b>wncasecmp</b>	<b>wsncat</b>
<b>wsncmp</b>	<b>wsncpy</b>
<b>wspbrk</b>	<b>wsprintf</b>
<b>wsrchr</b>	<b>wsscancf</b>
<b>wssp</b>	<b>wstod</b>
<b>wstok</b>	<b>wstol</b>
<b>wstoll</b>	<b>wstostr</b>
<b>wsxfrm</b>	<b>_xftw64</b>
<b>__xpg4_putmsg</b>	<b>__xpg4_putpmsg</b>

**FILES**

**/usr/lib/libc.so.1**  
**/usr/lib/libc.a**

shared object  
archive library

**ATTRIBUTES**

See **attributes(5)** for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsu
MT-Level	Safe

**SEE ALSO**

**pvs(1), intro(2), intro(3), intro(4), attributes(5), interface64(5), standards(5)**



**NAME** libci – Sun Solstice Enterprise Agent Component Interface Library

**SYNOPSIS** `cc [ flag ... ] file ... -lci -ldmi -lnsl -lrwtool [ library .. ]`

**DESCRIPTION** The **libci** library provides Component Interface API functions.

**INTERFACES** **DmiRegisterCi**                    **DmiUnRegisterCi**                    **DmiOriginateEvent**

**ATTRIBUTES** See **attributes(5)** for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT-Level	Unsafe

**SEE ALSO** **libdmi(4)**, **attributes(5)**

<b>NAME</b>	libdevid – device id library									
<b>SYNOPSIS</b>	cc [ <i>flag ...</i> ] <i>file ...</i> -ldevid [ <i>library ...</i> ] #include <devid.h>									
<b>DESCRIPTION</b>	<p>Functions in this library provide unique device ids for identifying a device, independent of the device's name or device number.</p> <p>The shared object <b>libdevid.so.1</b> provides the public interfaces defined below.</p> <p>For additional information on shared object interfaces, see <b>intro(4)</b>.</p>									
<b>INTERFACES</b>	<p>SUNW_1.1 (global):</p> <table> <tr> <td><b>devid_compare</b></td> <td><b>devid_deviceid_to_nmlist</b></td> </tr> <tr> <td><b>devid_free</b></td> <td><b>devid_free_nmlist</b></td> </tr> <tr> <td><b>devid_get</b></td> <td><b>devid_get_minor_name</b></td> </tr> <tr> <td><b>devid_sizeof</b></td> <td></td> </tr> </table>		<b>devid_compare</b>	<b>devid_deviceid_to_nmlist</b>	<b>devid_free</b>	<b>devid_free_nmlist</b>	<b>devid_get</b>	<b>devid_get_minor_name</b>	<b>devid_sizeof</b>	
<b>devid_compare</b>	<b>devid_deviceid_to_nmlist</b>									
<b>devid_free</b>	<b>devid_free_nmlist</b>									
<b>devid_get</b>	<b>devid_get_minor_name</b>									
<b>devid_sizeof</b>										
<b>FILES</b>	/usr/lib/libdevid.so.1	The location of the device id library interfaces.								
	/usr/lib/libdevid.so	A symlink to /usr/lib/libdevid.so.1.								
<b>ATTRIBUTES</b>	See <b>attributes(5)</b> for description of the following attributes:									
	<table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>MT Level</td> <td>MT-Safe</td> </tr> </tbody> </table>		ATTRIBUTE TYPE	ATTRIBUTE VALUE	MT Level	MT-Safe				
ATTRIBUTE TYPE	ATTRIBUTE VALUE									
MT Level	MT-Safe									
<b>SEE ALSO</b>	pvs(1), intro(4), attributes(5)									

<b>NAME</b>	libdl – the dynamic linking interface library				
<b>SYNOPSIS</b>	<code>cc [ flag ... ] file ... -ldl [ library ... ]</code>				
<b>DESCRIPTION</b>	<p>Functions in this library provide direct access to the dynamic linking facilities. This library is implemented as a <i>filter</i> on the runtime linker (see <b>ld.so.1(1)</b>).</p> <p>The shared object <b>libdl.so.1</b> provides the public interfaces defined below.</p> <p>For additional information on shared object interfaces, see <b>intro(4)</b>.</p>				
<b>INTERFACES</b>	<p><b>SISCD_2.3 (SPARC only)</b> - The SPARC Compliance Definition, revision 2.3:</p> <p style="padding-left: 2em;"><b>dlclose</b>      <b>dlderror</b>      <b>dlopen</b>      <b>dlsym</b></p> <p><b>SUNW_1.1 (generic)</b> -</p> <p style="padding-left: 2em;"><b>dladdr</b></p> <p><b>SUNW_1.2 (generic)</b> - This interface inherits all definitions from SUNW_1.1 and defines:</p> <p style="padding-left: 2em;"><b>dldump</b></p> <p><b>SUNW_1.3 (generic)</b> - This interface inherits all definitions from SUNW_1.2 and defines:</p> <p style="padding-left: 2em;"><b>dlinfo</b>                      <b>dldmopen</b></p> <p><b>SUNW_1.1 (SPARC)</b> - This interface inherits all definitions from SISCD_2.3.</p> <p><b>SUNW_1.1 (i386)</b> - This interface contains all SISCD_2.3 definitions.</p>				
<b>FILES</b>	<table border="0" style="width: 100%;"> <tr> <td style="width: 50%;"><code>/usr/lib/libdl.so.1</code></td> <td>shared object</td> </tr> <tr> <td><code>/etc/lib/libdl.so.1</code></td> <td>shared object (copy)</td> </tr> </table>	<code>/usr/lib/libdl.so.1</code>	shared object	<code>/etc/lib/libdl.so.1</code>	shared object (copy)
<code>/usr/lib/libdl.so.1</code>	shared object				
<code>/etc/lib/libdl.so.1</code>	shared object (copy)				
<b>ATTRIBUTES</b>	See <b>attributes(5)</b> for descriptions of the following attributes:				
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">ATTRIBUTE TYPE</th> <th style="text-align: left;">ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>MT Level</td> <td>Safe</td> </tr> </tbody> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	MT Level	Safe
ATTRIBUTE TYPE	ATTRIBUTE VALUE				
MT Level	Safe				
<b>SEE ALSO</b>	<b>ld.so.1(1)</b> , <b>pvs(1)</b> , <b>intro(4)</b> , <b>attributes(5)</b>				

<b>NAME</b>	libdmi – Sun Solstice Enterprise Agent DMI Library
<b>DESCRIPTION</b>	The <b>libdmi</b> library is a Solstice Enterprise Agent DMI generic library. It supports the DMI service provider, management application, and component instrumentation with data encoding, RPC communication, and other functionalities. This library is linked with management application and component instrumentation programs.
<b>SEE ALSO</b>	<b>libci(4)</b> , <b>libdmimi(4)</b>

<b>NAME</b>	libdmimi – Sun Solstice Enterprise Agent Management Interface Library																								
<b>SYNOPSIS</b>	<code>cc [ <i>flag</i> ... ] <i>file</i> ... -ldmimi -ldmi -lnsl -lrwtool [ <i>library</i> .. ]</code>																								
<b>DESCRIPTION</b>	The <b>libdmimi</b> library provides Management Interface API functions.																								
<b>INTERFACES</b>	<p>Initialization functions:</p> <table border="0"> <tr> <td><b>DmiGetConfig</b></td> <td><b>DmiGetVersion</b></td> <td><b>DmiRegister</b></td> </tr> <tr> <td><b>DmiSetConfig</b></td> <td><b>DmiUnregister</b></td> <td></td> </tr> </table> <p>Listing functions:</p> <table border="0"> <tr> <td><b>DmiListAttributes</b></td> <td><b>DmiListClassNames</b></td> <td><b>DmiListComponents</b></td> </tr> <tr> <td><b>DmiListComponentsByClass</b></td> <td><b>DmiListGroups</b></td> <td><b>DmiListLanguages</b></td> </tr> </table> <p>Operation functions:</p> <table border="0"> <tr> <td><b>DmiAddRow</b></td> <td><b>DmiDeleteRow</b></td> <td><b>DmiGetAttributes</b></td> </tr> <tr> <td><b>DmiGetMultiple</b></td> <td><b>DmiSetAttributes</b></td> <td><b>DmiSetMultiple</b></td> </tr> </table> <p>Data administration functions:</p> <table border="0"> <tr> <td><b>DmiAddComponent</b></td> <td><b>DmiAddGroup</b></td> <td><b>DmiAddLanguage</b></td> </tr> <tr> <td><b>DmiDeleteComponent</b></td> <td><b>DmiDeleteGroup</b></td> <td><b>DmiDeleteLanguage</b></td> </tr> </table>	<b>DmiGetConfig</b>	<b>DmiGetVersion</b>	<b>DmiRegister</b>	<b>DmiSetConfig</b>	<b>DmiUnregister</b>		<b>DmiListAttributes</b>	<b>DmiListClassNames</b>	<b>DmiListComponents</b>	<b>DmiListComponentsByClass</b>	<b>DmiListGroups</b>	<b>DmiListLanguages</b>	<b>DmiAddRow</b>	<b>DmiDeleteRow</b>	<b>DmiGetAttributes</b>	<b>DmiGetMultiple</b>	<b>DmiSetAttributes</b>	<b>DmiSetMultiple</b>	<b>DmiAddComponent</b>	<b>DmiAddGroup</b>	<b>DmiAddLanguage</b>	<b>DmiDeleteComponent</b>	<b>DmiDeleteGroup</b>	<b>DmiDeleteLanguage</b>
<b>DmiGetConfig</b>	<b>DmiGetVersion</b>	<b>DmiRegister</b>																							
<b>DmiSetConfig</b>	<b>DmiUnregister</b>																								
<b>DmiListAttributes</b>	<b>DmiListClassNames</b>	<b>DmiListComponents</b>																							
<b>DmiListComponentsByClass</b>	<b>DmiListGroups</b>	<b>DmiListLanguages</b>																							
<b>DmiAddRow</b>	<b>DmiDeleteRow</b>	<b>DmiGetAttributes</b>																							
<b>DmiGetMultiple</b>	<b>DmiSetAttributes</b>	<b>DmiSetMultiple</b>																							
<b>DmiAddComponent</b>	<b>DmiAddGroup</b>	<b>DmiAddLanguage</b>																							
<b>DmiDeleteComponent</b>	<b>DmiDeleteGroup</b>	<b>DmiDeleteLanguage</b>																							
<b>ATTRIBUTES</b>	<p>See <b>attributes(5)</b> for descriptions of the following attributes:</p> <table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>MT-Level</td> <td>Unsafe</td> </tr> </tbody> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	MT-Level	Unsafe																				
ATTRIBUTE TYPE	ATTRIBUTE VALUE																								
MT-Level	Unsafe																								
<b>SEE ALSO</b>	<b>libdmi(4)</b> , <b>attributes(5)</b>																								

<b>NAME</b>	libelf – ELF access library																																										
<b>SYNOPSIS</b>	<code>cc [ <i>flag</i> ... ] <i>file</i> ... -lelf [ <i>library</i> ... ]</code> <code>#include &lt;libelf.h&gt;</code>																																										
<b>DESCRIPTION</b>	<p>Functions in this library let a program manipulate ELF (Executable and Linking Format) object files, archive files, and archive members. The header provides type and function declarations for all library services.</p> <p>The shared object <b>libelf.so.1</b> provides the public interfaces defined below.</p> <p>For additional information on shared object interfaces, see <b>intro(4)</b>.</p>																																										
<b>INTERFACES</b>	<p>SUNW_1.1 (generic):</p> <table border="0"> <tr> <td><code>elf32_fsize</code></td> <td><code>elf32_getehdr</code></td> <td><code>elf32_getphdr</code></td> </tr> <tr> <td><code>elf32_getshdr</code></td> <td><code>elf32_newehdr</code></td> <td><code>elf32_newphdr</code></td> </tr> <tr> <td><code>elf32_xlatetof</code></td> <td><code>elf32_xlatetom</code></td> <td><code>elf_begin</code></td> </tr> <tr> <td><code>elf_cntl</code></td> <td><code>elf_end</code></td> <td><code>elf_errmsg</code></td> </tr> <tr> <td><code>elf_errno</code></td> <td><code>elf_fill</code></td> <td><code>elf_flagdata</code></td> </tr> <tr> <td><code>elf_flagehdr</code></td> <td><code>elf_flagelf</code></td> <td><code>elf_flagphdr</code></td> </tr> <tr> <td><code>elf_flagscn</code></td> <td><code>elf_flagshdr</code></td> <td><code>elf_getarhdr</code></td> </tr> <tr> <td><code>elf_getarsym</code></td> <td><code>elf_getbase</code></td> <td><code>elf_getdata</code></td> </tr> <tr> <td><code>elf_getident</code></td> <td><code>elf_getscn</code></td> <td><code>elf_hash</code></td> </tr> <tr> <td><code>elf_kind</code></td> <td><code>elf_memory</code></td> <td><code>elf_ndxscn</code></td> </tr> <tr> <td><code>elf_newdata</code></td> <td><code>elf_newscn</code></td> <td><code>elf_next</code></td> </tr> <tr> <td><code>elf_nextscn</code></td> <td><code>elf_rand</code></td> <td><code>elf_rawdata</code></td> </tr> <tr> <td><code>elf_rawfile</code></td> <td><code>elf_strptr</code></td> <td><code>elf_update</code></td> </tr> <tr> <td><code>elf_version</code></td> <td><code>nlist</code></td> <td></td> </tr> </table>	<code>elf32_fsize</code>	<code>elf32_getehdr</code>	<code>elf32_getphdr</code>	<code>elf32_getshdr</code>	<code>elf32_newehdr</code>	<code>elf32_newphdr</code>	<code>elf32_xlatetof</code>	<code>elf32_xlatetom</code>	<code>elf_begin</code>	<code>elf_cntl</code>	<code>elf_end</code>	<code>elf_errmsg</code>	<code>elf_errno</code>	<code>elf_fill</code>	<code>elf_flagdata</code>	<code>elf_flagehdr</code>	<code>elf_flagelf</code>	<code>elf_flagphdr</code>	<code>elf_flagscn</code>	<code>elf_flagshdr</code>	<code>elf_getarhdr</code>	<code>elf_getarsym</code>	<code>elf_getbase</code>	<code>elf_getdata</code>	<code>elf_getident</code>	<code>elf_getscn</code>	<code>elf_hash</code>	<code>elf_kind</code>	<code>elf_memory</code>	<code>elf_ndxscn</code>	<code>elf_newdata</code>	<code>elf_newscn</code>	<code>elf_next</code>	<code>elf_nextscn</code>	<code>elf_rand</code>	<code>elf_rawdata</code>	<code>elf_rawfile</code>	<code>elf_strptr</code>	<code>elf_update</code>	<code>elf_version</code>	<code>nlist</code>	
<code>elf32_fsize</code>	<code>elf32_getehdr</code>	<code>elf32_getphdr</code>																																									
<code>elf32_getshdr</code>	<code>elf32_newehdr</code>	<code>elf32_newphdr</code>																																									
<code>elf32_xlatetof</code>	<code>elf32_xlatetom</code>	<code>elf_begin</code>																																									
<code>elf_cntl</code>	<code>elf_end</code>	<code>elf_errmsg</code>																																									
<code>elf_errno</code>	<code>elf_fill</code>	<code>elf_flagdata</code>																																									
<code>elf_flagehdr</code>	<code>elf_flagelf</code>	<code>elf_flagphdr</code>																																									
<code>elf_flagscn</code>	<code>elf_flagshdr</code>	<code>elf_getarhdr</code>																																									
<code>elf_getarsym</code>	<code>elf_getbase</code>	<code>elf_getdata</code>																																									
<code>elf_getident</code>	<code>elf_getscn</code>	<code>elf_hash</code>																																									
<code>elf_kind</code>	<code>elf_memory</code>	<code>elf_ndxscn</code>																																									
<code>elf_newdata</code>	<code>elf_newscn</code>	<code>elf_next</code>																																									
<code>elf_nextscn</code>	<code>elf_rand</code>	<code>elf_rawdata</code>																																									
<code>elf_rawfile</code>	<code>elf_strptr</code>	<code>elf_update</code>																																									
<code>elf_version</code>	<code>nlist</code>																																										
<b>FILES</b>	<table border="0"> <tr> <td><code>/usr/lib/libelf.so.1</code></td> <td>shared object</td> </tr> <tr> <td><code>/usr/lib/libelf.a</code></td> <td>archive library</td> </tr> </table>	<code>/usr/lib/libelf.so.1</code>	shared object	<code>/usr/lib/libelf.a</code>	archive library																																						
<code>/usr/lib/libelf.so.1</code>	shared object																																										
<code>/usr/lib/libelf.a</code>	archive library																																										
<b>ATTRIBUTES</b>	See <b>attributes(5)</b> for descriptions of the following attributes:																																										
	<table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>MT-Level</td> <td>Safe</td> </tr> </tbody> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	MT-Level	Safe																																						
ATTRIBUTE TYPE	ATTRIBUTE VALUE																																										
MT-Level	Safe																																										
<b>SEE ALSO</b>	<b>pvs(1)</b> , <b>elf(3E)</b> , <b>intro(4)</b> , <b>attributes(5)</b>																																										

**NAME** libintl – internationalization library

**SYNOPSIS** `cc [ flag ... ] file ... -lintl [ library ... ]`

`#include <libintl.h>`

`#include <locale.h> /* needed for dcgettext() only */`

**DESCRIPTION** Historically, functions in this library provided wide character translations. This functionality now resides in **libc(4)**.

This library is maintained to provide backward compatibility for both runtime and compilation environments. The shared object version is implemented as a filter on **libintl.so.1**, and the archive version is implemented as a null archive. New application development need not reference either version of **libintl**.

The shared object **libintl.so.1** provides the public interfaces defined below.

For additional information on shared object interfaces, see **intro(4)**.

**INTERFACES** SUNW\_1.1 (generic):

**bindtextdomain**  
**gettext**

**dcgettext**  
**textdomain**

**dgettext**

**FILES** `/usr/lib/libintl.so.1`

a filter on **libc.so.1**

`/usr/lib/libintl.a`

a link to `/usr/lib/null.a`

**ATTRIBUTES**

See **attributes(5)** for descriptions of the following attributes:

`/usr/lib/libintl.so.1TT`

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsu
MT-Level	Safe with exceptions

**SEE ALSO**

**pvs(1)**, **gettext(3C)**, **intro(4)**, **libc(4)**, **attributes(5)**

<b>NAME</b>	libkrb – Kerberos library																																										
<b>SYNOPSIS</b>	cc [ <i>flag ...</i> ] <i>file ...</i> -lkrb [ <i>library ...</i> ] #include <kerberos/krb.h> #include <netinet/in.h>																																										
<b>DESCRIPTION</b>	Functions in this library provide Kerberos utility routines. The shared object <b>libkrb.so.1</b> provides the public interfaces defined below. For additional information on shared object interfaces, see <b>intro(4)</b> .																																										
<b>INTERFACES</b>	<p>SUNW_1.1 (generic):</p> <table border="0"> <tr> <td>ErrorMsg</td> <td>LineNbr</td> <td>authkerb_create</td> </tr> <tr> <td>authkerb_getucred</td> <td>authkerb_seccreate</td> <td>create_auth_reply</td> </tr> <tr> <td>error_table_name</td> <td>_et_list</td> <td>kerb_error</td> </tr> <tr> <td>kerb_get_session_cred</td> <td>kerb_get_session_key</td> <td>klog</td> </tr> <tr> <td>_kmsgout</td> <td>krbONE</td> <td>krb_err_txt</td> </tr> <tr> <td>krb_get_admhst</td> <td>krb_get_cred</td> <td>krb_get_default_realm</td> </tr> <tr> <td>krb_get_krbhst</td> <td>krb_get_lrealm</td> <td>krb_get_phost</td> </tr> <tr> <td>krb_kntoln</td> <td>krb_mk_err</td> <td>krb_mk_req</td> </tr> <tr> <td>krb_mk_safe</td> <td>krb_net_read</td> <td>krb_net_write</td> </tr> <tr> <td>krb_rd_err</td> <td>krb_rd_req</td> <td>krb_rd_safe</td> </tr> <tr> <td>krb_realmofhost</td> <td>krb_rcvauth</td> <td>krb_sendauth</td> </tr> <tr> <td>krb_set_key</td> <td>krb_set_tkt_string</td> <td>log</td> </tr> <tr> <td>pkt_cipher</td> <td>_svcauth_kerb</td> <td>svc_kerb_reg</td> </tr> <tr> <td>tkt_string</td> <td>xdr_authkerb_cred</td> <td>xdr_authkerb_verf</td> </tr> </table>	ErrorMsg	LineNbr	authkerb_create	authkerb_getucred	authkerb_seccreate	create_auth_reply	error_table_name	_et_list	kerb_error	kerb_get_session_cred	kerb_get_session_key	klog	_kmsgout	krbONE	krb_err_txt	krb_get_admhst	krb_get_cred	krb_get_default_realm	krb_get_krbhst	krb_get_lrealm	krb_get_phost	krb_kntoln	krb_mk_err	krb_mk_req	krb_mk_safe	krb_net_read	krb_net_write	krb_rd_err	krb_rd_req	krb_rd_safe	krb_realmofhost	krb_rcvauth	krb_sendauth	krb_set_key	krb_set_tkt_string	log	pkt_cipher	_svcauth_kerb	svc_kerb_reg	tkt_string	xdr_authkerb_cred	xdr_authkerb_verf
ErrorMsg	LineNbr	authkerb_create																																									
authkerb_getucred	authkerb_seccreate	create_auth_reply																																									
error_table_name	_et_list	kerb_error																																									
kerb_get_session_cred	kerb_get_session_key	klog																																									
_kmsgout	krbONE	krb_err_txt																																									
krb_get_admhst	krb_get_cred	krb_get_default_realm																																									
krb_get_krbhst	krb_get_lrealm	krb_get_phost																																									
krb_kntoln	krb_mk_err	krb_mk_req																																									
krb_mk_safe	krb_net_read	krb_net_write																																									
krb_rd_err	krb_rd_req	krb_rd_safe																																									
krb_realmofhost	krb_rcvauth	krb_sendauth																																									
krb_set_key	krb_set_tkt_string	log																																									
pkt_cipher	_svcauth_kerb	svc_kerb_reg																																									
tkt_string	xdr_authkerb_cred	xdr_authkerb_verf																																									
<b>FILES</b>	<table border="0"> <tr> <td>/usr/lib/libkrb.so.1</td> <td>shared object</td> </tr> <tr> <td>/usr/lib/libkrb.a</td> <td>archive library</td> </tr> </table>	/usr/lib/libkrb.so.1	shared object	/usr/lib/libkrb.a	archive library																																						
/usr/lib/libkrb.so.1	shared object																																										
/usr/lib/libkrb.a	archive library																																										
<b>ATTRIBUTES</b>	See <b>attributes(5)</b> for descriptions of the following attributes:																																										
	<table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>MT-Level</td> <td>Unsafe</td> </tr> </tbody> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	MT-Level	Unsafe																																						
ATTRIBUTE TYPE	ATTRIBUTE VALUE																																										
MT-Level	Unsafe																																										
<b>SEE ALSO</b>	<b>pvs(1)</b> , <b>kerberos(3N)</b> , <b>intro(4)</b> , <b>attributes(5)</b>																																										



<b>NAME</b>	libkstat – kernel statistics library									
<b>SYNOPSIS</b>	<code>cc [ <i>flag</i> ... ] <i>file</i> ... -lkstat [ <i>library</i> ... ]</code> <code>#include &lt;kstat.h&gt;</code>									
<b>DESCRIPTION</b>	<p>Functions in this library provide a general-purpose mechanism for providing kernel statistics to users.</p> <p>The shared object <b>libkstat.so.1</b> provides the public interfaces defined below.</p> <p>For additional information on shared object interfaces, see <b>intro(4)</b>.</p>									
<b>INTERFACES</b>	<p>SUNW_1.1 (generic):</p> <table> <tr> <td><code>kstat_chain_update</code></td> <td><code>kstat_close</code></td> <td><code>kstat_data_lookup</code></td> </tr> <tr> <td><code>kstat_lookup</code></td> <td><code>kstat_open</code></td> <td><code>kstat_read</code></td> </tr> <tr> <td><code>kstat_write</code></td> <td></td> <td></td> </tr> </table>	<code>kstat_chain_update</code>	<code>kstat_close</code>	<code>kstat_data_lookup</code>	<code>kstat_lookup</code>	<code>kstat_open</code>	<code>kstat_read</code>	<code>kstat_write</code>		
<code>kstat_chain_update</code>	<code>kstat_close</code>	<code>kstat_data_lookup</code>								
<code>kstat_lookup</code>	<code>kstat_open</code>	<code>kstat_read</code>								
<code>kstat_write</code>										
<b>FILES</b>	<code>/usr/lib/libkstat.so.1</code> shared object									
<b>ATTRIBUTES</b>	See <b>attributes(5)</b> for descriptions of the following attributes:									
	<table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>MT-Level</td> <td>Unsafe</td> </tr> </tbody> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	MT-Level	Unsafe					
ATTRIBUTE TYPE	ATTRIBUTE VALUE									
MT-Level	Unsafe									
<b>SEE ALSO</b>	<b>pvs(1)</b> , <b>kstat(3K)</b> , <b>intro(4)</b> , <b>attributes(5)</b>									

**NAME** libkvm – Kernel Virtual Memory access library

**SYNOPSIS** `cc [ flag ... ] file ... -lkvm [ library ... ]`  
`#include <kvm.h>`

**DESCRIPTION** Functions in this library provide application access to kernel symbols, addresses and values. The individual routines are documented in Section 3K of the reference manuals. All of the **libkvm** routines are UNCOMMITTED. The UNCOMMITTED classification is due to the fact that there is almost nothing which can be put as a symbol in a namelist which has release-to-release stability. The syntax of these routines is historically stable release-to-release, but being UNCOMMITTED, the door is always open for change. The shared object **libkvm.so.1** provides the public interfaces defined below. For additional information on shared object interfaces, see **intro(4)**.

**INTERFACES** SUNW\_1.1 (generic):

<code>kvm_close</code>	<code>kvm_getcmd</code>	<code>kvm_getproc</code>
<code>kvm_getu</code>	<code>kvm_kread</code>	<code>kvm_kwrite</code>
<code>kvm_nextproc</code>	<code>kvm_nlist</code>	<code>kvm_open</code>
<code>kvm_read</code>	<code>kvm_setproc</code>	<code>kvm_uread</code>
<code>kvm_uwrite</code>	<code>kvm_write</code>	

**FILES** `/usr/lib/libkvm.so.1` shared object

**ATTRIBUTES** See **attributes(5)** for descriptions of the following attributes:

`/usr/lib/libkvm.so.1`

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWkvm
MT-Level	Unsafe

**SEE ALSO** **pvs(1)**, **intro(4)**, **attributes(5)**

<b>NAME</b>	libmapmalloc – an alternative memory allocator library									
<b>SYNOPSIS</b>	<b>cc</b> [ <i>flag</i> ... ] <i>file</i> ... <b>-lmapmalloc</b> [ <i>library</i> ... ] <b>#include</b> <stdlib.h>									
<b>DESCRIPTION</b>	Functions in this library provide a collection of <b>malloc</b> routines that use <b>mmap</b> (2) instead of <b>sbrk</b> (2) for acquiring heap space. The shared object <b>libmapmalloc.so.1</b> provides the public interfaces defined below. For additional information on shared object interfaces, see <b>intro</b> (4).									
<b>INTERFACES</b>	<b>SUNW_1.1</b> (generic): <table> <tr> <td><b>calloc</b></td> <td><b>cfree</b></td> <td><b>free</b></td> </tr> <tr> <td><b>mallinfo</b></td> <td><b>malloc</b></td> <td><b>mallopt</b></td> </tr> <tr> <td><b>memalign</b></td> <td><b>realloc</b></td> <td><b>valloc</b></td> </tr> </table>	<b>calloc</b>	<b>cfree</b>	<b>free</b>	<b>mallinfo</b>	<b>malloc</b>	<b>mallopt</b>	<b>memalign</b>	<b>realloc</b>	<b>valloc</b>
<b>calloc</b>	<b>cfree</b>	<b>free</b>								
<b>mallinfo</b>	<b>malloc</b>	<b>mallopt</b>								
<b>memalign</b>	<b>realloc</b>	<b>valloc</b>								
<b>FILES</b>	<b>/usr/lib/libmapmalloc.so.1</b> shared object <b>/usr/lib/libmapmalloc.a</b> archive library.									
<b>ATTRIBUTES</b>	See <b>attributes</b> (5) for descriptions of the following attributes: <table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>MT-Level</td> <td>Safe</td> </tr> </tbody> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	MT-Level	Safe					
ATTRIBUTE TYPE	ATTRIBUTE VALUE									
MT-Level	Safe									
<b>SEE ALSO</b>	<b>pvs</b> (1), <b>mmap</b> (2), <b>sbrk</b> (2), <b>malloc</b> (3C), <b>malloc</b> (3X), <b>mapmalloc</b> (3X), <b>intro</b> (4), <b>attributes</b> (5)									

<b>NAME</b>	libmp – multiple precision library																		
<b>SYNOPSIS</b>	<code>cc [ <i>flag</i> ... ] <i>file</i> ... -l<b>mp</b> [ <i>library</i> ... ]</code> <code>#include &lt;mp.h&gt;</code>																		
<b>DESCRIPTION</b>	<p>Functions in this library provide various multiple precision routines.</p> <p>The shared object <b>libmp.so.2</b> provides the public interfaces defined below. See <b>INTERFACES</b>.</p> <p>The shared object <b>libmp.so.1</b> is available for backwards compatibility purposes and provides the older versions of these interfaces without the <b>mp_</b> prepended to them.</p> <p>Care should be taken in using the static version of this library <b>libmp.a</b> because it contains both the current and old interfaces.</p> <p>For additional information on shared object interfaces, see <b>intro(4)</b>.</p>																		
<b>INTERFACES</b>	<p>SUNW_1.1 (generic):</p> <table border="0"> <tr> <td><b>mp_gcd</b></td> <td><b>mp_itom</b></td> <td><b>mp_madd</b></td> </tr> <tr> <td><b>mp_mcmp</b></td> <td><b>mp_mdiv</b></td> <td><b>mp_mfree</b></td> </tr> <tr> <td><b>mp_min</b></td> <td><b>mp_mout</b></td> <td><b>mp_msqrt</b></td> </tr> <tr> <td><b>mp_msub</b></td> <td><b>mp_mtox</b></td> <td><b>mp_mult</b></td> </tr> <tr> <td><b>mp_pow</b></td> <td><b>mp_rpow</b></td> <td><b>mp_sdiv</b></td> </tr> <tr> <td><b>mp_xtom</b></td> <td></td> <td></td> </tr> </table>	<b>mp_gcd</b>	<b>mp_itom</b>	<b>mp_madd</b>	<b>mp_mcmp</b>	<b>mp_mdiv</b>	<b>mp_mfree</b>	<b>mp_min</b>	<b>mp_mout</b>	<b>mp_msqrt</b>	<b>mp_msub</b>	<b>mp_mtox</b>	<b>mp_mult</b>	<b>mp_pow</b>	<b>mp_rpow</b>	<b>mp_sdiv</b>	<b>mp_xtom</b>		
<b>mp_gcd</b>	<b>mp_itom</b>	<b>mp_madd</b>																	
<b>mp_mcmp</b>	<b>mp_mdiv</b>	<b>mp_mfree</b>																	
<b>mp_min</b>	<b>mp_mout</b>	<b>mp_msqrt</b>																	
<b>mp_msub</b>	<b>mp_mtox</b>	<b>mp_mult</b>																	
<b>mp_pow</b>	<b>mp_rpow</b>	<b>mp_sdiv</b>																	
<b>mp_xtom</b>																			
<b>FILES</b>	<p><b>/usr/lib/libmp.so.1</b> shared object file available for backwards compatibility purposes</p> <p><b>/usr/lib/libmp.so.2</b> shared object file</p> <p><b>/usr/lib/libmp.so</b> archive library</p>																		
<b>ATTRIBUTES</b>	<p>See <b>attributes(5)</b> for descriptions of the following attributes:</p> <table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>MT-Level</td> <td>Unsafe</td> </tr> </tbody> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	MT-Level	Unsafe														
ATTRIBUTE TYPE	ATTRIBUTE VALUE																		
MT-Level	Unsafe																		
<b>SEE ALSO</b>	<b>pvs(1)</b> , <b>exp(3M)</b> , <b>mp(3M)</b> , <b>intro(4)</b> , <b>attributes(5)</b>																		

<b>NAME</b>	libnisdb – NIS+ Database access library													
<b>SYNOPSIS</b>	<pre>cc [ flag ... ] file ... -lnisdb -lnsl [ library ... ] #include &lt;rpcsvc/nis.h&gt; #include &lt;rpcsvc/nis_db.h&gt;</pre>													
<b>DESCRIPTION</b>	<p>Functions in this library describe the interface between the NIS+ server and the underlying database.</p> <p>The shared object <b>libnisdb.so.2</b> provides the public interfaces defined below.</p> <p>For additional information on shared object interfaces, see <b>intro(4)</b>.</p>													
<b>INTERFACES</b>	<p><b>SUNW_2.1</b> (generic):</p> <table> <tr> <td><b>db_create_table</b></td> <td><b>db_destroy_table</b></td> <td><b>db_first_entry</b></td> </tr> <tr> <td><b>db_initialize</b></td> <td><b>db_list_entries</b></td> <td><b>db_message_dict</b></td> </tr> <tr> <td><b>db_next_entry</b></td> <td><b>db_remove_entry</b></td> <td><b>db_reset_next_entry</b></td> </tr> <tr> <td><b>db_standby</b></td> <td><b>db_table_exists</b></td> <td><b>db_unload_table</b></td> </tr> </table>		<b>db_create_table</b>	<b>db_destroy_table</b>	<b>db_first_entry</b>	<b>db_initialize</b>	<b>db_list_entries</b>	<b>db_message_dict</b>	<b>db_next_entry</b>	<b>db_remove_entry</b>	<b>db_reset_next_entry</b>	<b>db_standby</b>	<b>db_table_exists</b>	<b>db_unload_table</b>
<b>db_create_table</b>	<b>db_destroy_table</b>	<b>db_first_entry</b>												
<b>db_initialize</b>	<b>db_list_entries</b>	<b>db_message_dict</b>												
<b>db_next_entry</b>	<b>db_remove_entry</b>	<b>db_reset_next_entry</b>												
<b>db_standby</b>	<b>db_table_exists</b>	<b>db_unload_table</b>												
<b>FILES</b>	<pre>/usr/lib/libnisdb.so.2</pre>	shared object												
	<pre>/usr/lib/libnisdb.a</pre>	archive library												
<b>ATTRIBUTES</b>	See <b>attributes(5)</b> for descriptions of the following attributes:													
	<table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>MT-Level</td> <td>Unsafe</td> </tr> </tbody> </table>		ATTRIBUTE TYPE	ATTRIBUTE VALUE	MT-Level	Unsafe								
ATTRIBUTE TYPE	ATTRIBUTE VALUE													
MT-Level	Unsafe													
<b>SEE ALSO</b>	<b>pvs(1)</b> , <b>nis_db(3N)</b> , <b>intro(4)</b> , <b>attributes(5)</b>													

<b>NAME</b>	libnsl – the network services library																																																												
<b>SYNOPSIS</b>	<code>cc [ <i>flag</i> ... ] <i>file</i> ... -lnsl [ <i>library</i> ... ]</code>																																																												
<b>DESCRIPTION</b>	<p>Functions in this library provide routines that provide a transport-level interface to networking services for applications, facilities for machine-independent data representation, a remote procedure call mechanism, and other networking services useful for application programs.</p> <p>The shared object <b>libnsl.so.1</b> provides the public interfaces defined below.</p> <p>For additional information on shared object interfaces, see <b>intro(4)</b>.</p> <p>Many features in this library are implemented upon dynamic linking.</p>																																																												
<b>INTERFACES</b>	<p><b>SYSVABI_1.3(generic)</b> - The System V Application Binary Interface, Third Edition:</p> <table border="0"> <tr> <td><b>authdes_getcred</b></td> <td><b>authdes_seccreate</b></td> </tr> <tr> <td><b>authnone_create</b></td> <td><b>authsys_create</b></td> </tr> <tr> <td><b>authsys_create_default</b></td> <td><b>clnt_create</b></td> </tr> <tr> <td><b>clnt_dg_create</b></td> <td><b>clnt_pcreateerror</b></td> </tr> <tr> <td><b>clnt_perrno</b></td> <td><b>clnt_perror</b></td> </tr> <tr> <td><b>clnt_raw_create</b></td> <td><b>clnt_screateerror</b></td> </tr> <tr> <td><b>clnt_serrno</b></td> <td><b>clnt_sperror</b></td> </tr> <tr> <td><b>clnt_tli_create</b></td> <td><b>clnt_tp_create</b></td> </tr> <tr> <td><b>clnt_vc_create</b></td> <td><b>endnetconfig</b></td> </tr> <tr> <td><b>endnetpath</b></td> <td><b>freenetconfignt</b></td> </tr> <tr> <td><b>getnetconfig</b></td> <td><b>getnetconfignt</b></td> </tr> <tr> <td><b>getnetname</b></td> <td><b>getnetpath</b></td> </tr> <tr> <td><b>getpublickey</b></td> <td><b>getsecretkey</b></td> </tr> <tr> <td><b>host2netname</b></td> <td><b>key_decryptsession</b></td> </tr> <tr> <td><b>key_encryptsession</b></td> <td><b>key_gendes</b></td> </tr> <tr> <td><b>key_setsecret</b></td> <td><b>nc_perror</b></td> </tr> <tr> <td><b>_nderror</b></td> <td><b>netdir_free</b></td> </tr> <tr> <td><b>netdir_getbyaddr</b></td> <td><b>netdir_getbyname</b></td> </tr> <tr> <td><b>netdir_options</b></td> <td><b>netname2host</b></td> </tr> <tr> <td><b>netname2user</b></td> <td><b>rpcb_getaddr</b></td> </tr> <tr> <td><b>rpcb_getmaps</b></td> <td><b>rpcb_gettime</b></td> </tr> <tr> <td><b>rpcb_rmtcall</b></td> <td><b>rpc_broadcast</b></td> </tr> <tr> <td><b>rpcb_set</b></td> <td><b>rpcb_unset</b></td> </tr> <tr> <td><b>rpc_call</b></td> <td><b>rpc_createerr</b></td> </tr> <tr> <td><b>rpc_reg</b></td> <td><b>setnetconfig</b></td> </tr> <tr> <td><b>setnetpath</b></td> <td><b>svc_create</b></td> </tr> <tr> <td><b>svc_dg_create</b></td> <td><b>svcerr_auth</b></td> </tr> <tr> <td><b>svcerr_decode</b></td> <td><b>svcerr_noproc</b></td> </tr> <tr> <td><b>svcerr_noprogram</b></td> <td><b>svcerr_progvers</b></td> </tr> <tr> <td><b>svcerr_systemerr</b></td> <td><b>svcerr_weakauth</b></td> </tr> </table>	<b>authdes_getcred</b>	<b>authdes_seccreate</b>	<b>authnone_create</b>	<b>authsys_create</b>	<b>authsys_create_default</b>	<b>clnt_create</b>	<b>clnt_dg_create</b>	<b>clnt_pcreateerror</b>	<b>clnt_perrno</b>	<b>clnt_perror</b>	<b>clnt_raw_create</b>	<b>clnt_screateerror</b>	<b>clnt_serrno</b>	<b>clnt_sperror</b>	<b>clnt_tli_create</b>	<b>clnt_tp_create</b>	<b>clnt_vc_create</b>	<b>endnetconfig</b>	<b>endnetpath</b>	<b>freenetconfignt</b>	<b>getnetconfig</b>	<b>getnetconfignt</b>	<b>getnetname</b>	<b>getnetpath</b>	<b>getpublickey</b>	<b>getsecretkey</b>	<b>host2netname</b>	<b>key_decryptsession</b>	<b>key_encryptsession</b>	<b>key_gendes</b>	<b>key_setsecret</b>	<b>nc_perror</b>	<b>_nderror</b>	<b>netdir_free</b>	<b>netdir_getbyaddr</b>	<b>netdir_getbyname</b>	<b>netdir_options</b>	<b>netname2host</b>	<b>netname2user</b>	<b>rpcb_getaddr</b>	<b>rpcb_getmaps</b>	<b>rpcb_gettime</b>	<b>rpcb_rmtcall</b>	<b>rpc_broadcast</b>	<b>rpcb_set</b>	<b>rpcb_unset</b>	<b>rpc_call</b>	<b>rpc_createerr</b>	<b>rpc_reg</b>	<b>setnetconfig</b>	<b>setnetpath</b>	<b>svc_create</b>	<b>svc_dg_create</b>	<b>svcerr_auth</b>	<b>svcerr_decode</b>	<b>svcerr_noproc</b>	<b>svcerr_noprogram</b>	<b>svcerr_progvers</b>	<b>svcerr_systemerr</b>	<b>svcerr_weakauth</b>
<b>authdes_getcred</b>	<b>authdes_seccreate</b>																																																												
<b>authnone_create</b>	<b>authsys_create</b>																																																												
<b>authsys_create_default</b>	<b>clnt_create</b>																																																												
<b>clnt_dg_create</b>	<b>clnt_pcreateerror</b>																																																												
<b>clnt_perrno</b>	<b>clnt_perror</b>																																																												
<b>clnt_raw_create</b>	<b>clnt_screateerror</b>																																																												
<b>clnt_serrno</b>	<b>clnt_sperror</b>																																																												
<b>clnt_tli_create</b>	<b>clnt_tp_create</b>																																																												
<b>clnt_vc_create</b>	<b>endnetconfig</b>																																																												
<b>endnetpath</b>	<b>freenetconfignt</b>																																																												
<b>getnetconfig</b>	<b>getnetconfignt</b>																																																												
<b>getnetname</b>	<b>getnetpath</b>																																																												
<b>getpublickey</b>	<b>getsecretkey</b>																																																												
<b>host2netname</b>	<b>key_decryptsession</b>																																																												
<b>key_encryptsession</b>	<b>key_gendes</b>																																																												
<b>key_setsecret</b>	<b>nc_perror</b>																																																												
<b>_nderror</b>	<b>netdir_free</b>																																																												
<b>netdir_getbyaddr</b>	<b>netdir_getbyname</b>																																																												
<b>netdir_options</b>	<b>netname2host</b>																																																												
<b>netname2user</b>	<b>rpcb_getaddr</b>																																																												
<b>rpcb_getmaps</b>	<b>rpcb_gettime</b>																																																												
<b>rpcb_rmtcall</b>	<b>rpc_broadcast</b>																																																												
<b>rpcb_set</b>	<b>rpcb_unset</b>																																																												
<b>rpc_call</b>	<b>rpc_createerr</b>																																																												
<b>rpc_reg</b>	<b>setnetconfig</b>																																																												
<b>setnetpath</b>	<b>svc_create</b>																																																												
<b>svc_dg_create</b>	<b>svcerr_auth</b>																																																												
<b>svcerr_decode</b>	<b>svcerr_noproc</b>																																																												
<b>svcerr_noprogram</b>	<b>svcerr_progvers</b>																																																												
<b>svcerr_systemerr</b>	<b>svcerr_weakauth</b>																																																												

<b>svc_fd_create</b>	<b>svc_fds</b>
<b>svc_getreqset</b>	<b>svc_raw_create</b>
<b>svc_reg</b>	<b>svc_run</b>
<b>svc_sendreply</b>	<b>svc_tli_create</b>
<b>svc_tp_create</b>	<b>svc_unreg</b>
<b>svc_vc_create</b>	<b>t_accept</b>
<b>taddr2uaddr</b>	<b>t_alloc</b>
<b>t_bind</b>	<b>t_close</b>
<b>t_connect</b>	<b>t_errno</b>
<b>t_error</b>	<b>t_free</b>
<b>t_getinfo</b>	<b>t_getstate</b>
<b>t_listen</b>	<b>t_look</b>
<b>t_open</b>	<b>t_optmgmt</b>
<b>t_rcv</b>	<b>t_rcvconnect</b>
<b>t_rcvdis</b>	<b>t_rcvrel</b>
<b>t_rcvudata</b>	<b>t_rcvuderr</b>
<b>t_snd</b>	<b>t_snddis</b>
<b>t_sndrel</b>	<b>t_sndudata</b>
<b>t_sync</b>	<b>t_unbind</b>
<b>uaddr2taddr</b>	<b>user2netname</b>
<b>xdr_accepted_reply</b>	<b>xdr_array</b>
<b>xdr_authsys_parms</b>	<b>xdr_bool</b>
<b>xdr_bytes</b>	<b>xdr_callhdr</b>
<b>xdr_callmsg</b>	<b>xdr_char</b>
<b>xdr_double</b>	<b>xdr_enum</b>
<b>xdr_float</b>	<b>xdr_free</b>
<b>xdr_int</b>	<b>xdr_long</b>
<b>xdrmem_create</b>	<b>xdr_opaque</b>
<b>xdr_opaque_auth</b>	<b>xdr_pointer</b>
<b>xdrrec_create</b>	<b>xdrrec_eof</b>
<b>xdrrec_skiprecord</b>	<b>xdr_reference</b>
<b>xdr_rejected_reply</b>	<b>xdr_replymsg</b>
<b>xdr_short</b>	<b>xdrstdio_create</b>
<b>xdr_string</b>	<b>xdr_u_char</b>
<b>xdr_u_long</b>	<b>xdr_union</b>
<b>xdr_u_short</b>	<b>xdr_vector</b>
<b>xdr_void</b>	<b>xdr_wrapstring</b>
<b>xprt_register</b>	<b>xprt_unregister</b>

**SISCD\_2.3 (SPARC only)** - The SPARC Compliance Definition, revision 2.3. This interface inherits all definitions from SYSVABI\_1.3, and defines:

<b>gethostbyaddr</b>	<b>gethostbyname</b>
<b>inet_addr</b>	<b>inet_netof</b>
<b>inet_ntoa</b>	<b>_null_auth</b>
<b>rpc_broadcast_exp</b>	<b>svc_fdset</b>

## SUNW\_1.1 (generic):

<b>authdes_create</b>	<b>authdes_lock</b>
<b>auth_destroy</b>	<b>callrpc</b>
<b>clnt_broadcast</b>	<b>clnt_call</b>
<b>clnt_control</b>	<b>clnt_create_timed</b>
<b>clnt_create_vers</b>	<b>clnt_destroy</b>
<b>clnt_freeres</b>	<b>clnt_geterr</b>
<b>clntraw_create</b>	<b>clnttcp_create</b>
<b>clnt_tp_create_timed</b>	<b>clntudp_bufcreate</b>
<b>clntudp_create</b>	<b>dbmclose</b>
<b>dbminit</b>	<b>delete</b>
<b>des_setparity</b>	<b>dial</b>
<b>doconfig</b>	<b>endhostent</b>
<b>endrpcent</b>	<b>fetch</b>
<b>firstkey</b>	<b>gethostbyaddr_r</b>
<b>gethostbyname_r</b>	<b>gethostent</b>
<b>gethostent_r</b>	<b>get_myaddress</b>
<b>getrpcbyname</b>	<b>getrpcbyname_r</b>
<b>getrpcbynumber</b>	<b>getrpcbynumber_r</b>
<b>getrpcent</b>	<b>getrpcent_r</b>
<b>getrpcport</b>	<b>h_errno</b>
<b>inet_ntoa_r</b>	<b>key_secretkey_is_set</b>
<b>maxbno</b>	<b>nc_sperror</b>
<b>netdir_perror</b>	<b>netdir_sperror</b>
<b>nextkey</b>	<b>nis_add</b>
<b>nis_add_entry</b>	<b>nis_addmember</b>
<b>nis_cache_add_entry_1</b>	<b>nis_cache_read_coldstart_1</b>
<b>nis_cache_refresh_entry_1</b>	<b>nis_cache_remove_entry_1</b>
<b>nis_checkpoint</b>	<b>nis_clone_object</b>
<b>nis_creategroup</b>	<b>nis_data</b>
<b>nis_destroygroup</b>	<b>nis_destroy_object</b>
<b>nis_dir_cmp</b>	<b>nis_domain_of</b>
<b>nis_dump</b>	<b>nis_dumplog</b>
<b>nis_finddirectory</b>	<b>nis_find_item</b>
<b>nis_first_entry</b>	<b>nis_freenames</b>
<b>nis_free_request</b>	<b>nis_freeresult</b>
<b>nis_freeservlist</b>	<b>nis_freetags</b>
<b>nis_getnames</b>	<b>nis_get_request</b>
<b>nis_getservlist</b>	<b>nis_get_static_storage</b>
<b>nis_insert_item</b>	<b>nis_insert_name</b>
<b>nis_in_table</b>	<b>nis_ismember</b>
<b>nis_leaf_of</b>	<b>nis_leaf_of_r</b>
<b>nis_terror</b>	<b>nis_list</b>
<b>nis_local_directory</b>	<b>nis_local_group</b>
<b>nis_local_host</b>	<b>nis_local_principal</b>



<b>nis_lookup</b>	<b>nis_make_error</b>
<b>nis_make_rpchandle</b>	<b>nis_mkdir</b>
<b>nis_modify</b>	<b>nis_modify_entry</b>
<b>nis_name_of</b>	<b>nis_next_entry</b>
<b>nis_perror</b>	<b>nis_ping</b>
<b>nis_print_directory</b>	<b>nis_print_entry</b>
<b>nis_print_group</b>	<b>nis_print_group_entry</b>
<b>nis_print_link</b>	<b>nis_print_object</b>
<b>nis_print_rights</b>	<b>nis_print_table</b>
<b>nis_read_obj</b>	<b>nis_remove</b>
<b>nis_remove_entry</b>	<b>nis_remove_item</b>
<b>nis_removemember</b>	<b>nis_remove_name</b>
<b>nis_rmdir</b>	<b>nis_servstate</b>
<b>nis_sperrno</b>	<b>nis_sperror</b>
<b>nis_sperror_r</b>	<b>nis_stats</b>
<b>nis_verifygroup</b>	<b>nis_write_obj</b>
<b>pmap_getmaps</b>	<b>pmap_getport</b>
<b>pmap_rmtcall</b>	<b>pmap_set</b>
<b>pmap_unset</b>	<b>registerrpc</b>
<b>rpc_control</b>	<b>sethostent</b>
<b>setrpcnt</b>	<b>store</b>
<b>svc_auth_reg</b>	<b>svc_control</b>
<b>svc_destroy</b>	<b>svc_dg_enablecache</b>
<b>svc_done</b>	<b>svc_exit</b>
<b>svcfld_create</b>	<b>svc_freargs</b>
<b>svc_getargs</b>	<b>svc_getreq</b>
<b>svc_getreq_common</b>	<b>svc_getreq_poll</b>
<b>svc_getrpcaller</b>	<b>svccraw_create</b>
<b>svc_register</b>	<b>svctcp_create</b>
<b>svcupd_bufcreate</b>	<b>svcupd_create</b>
<b>svc_unregister</b>	<b>__t_errno</b>
<b>t_getname</b>	<b>t_nerr</b>
<b>t_strerror</b>	<b>undial</b>
<b>xdr_destroy</b>	<b>xdr_getpos</b>
<b>xdr_hyper</b>	<b>xdr_inline</b>
<b>xdr_longlong_t</b>	<b>xdr_quadruple</b>
<b>xdrrec_endofrecord</b>	<b>xdrrec_readbytes</b>
<b>xdr_setpos</b>	<b>xdr_sizeof</b>
<b>xdr_u_hyper</b>	<b>xdr_u_int</b>
<b>xdr_u_longlong_t</b>	<b>yp_all</b>
<b>yp_bind</b>	<b>yperr_string</b>
<b>yp_first</b>	<b>yp_get_default_domain</b>
<b>yp_master</b>	<b>yp_match</b>
<b>yp_next</b>	<b>yp_order</b>
<b>ypprot_err</b>	<b>yp_unbind</b>

**yp\_update**

**SUNW\_1.1 (SPARC)** - This interface inherits all definitions from the generic SUNW\_1.1 and the SISCD\_2.3.

**SUNW\_1.1 (i386)** - This interface contains all definitions from SISCD\_2.3, and inherits all definitions from the generic SUNW\_1.1 and the SYSVABI\_1.3.

**FILES**

**/usr/lib/libnsl.so.1** shared object  
**/usr/lib/libnsl.a** archive library

**ATTRIBUTES**

See **attributes(5)** for descriptions of the following attributes:

**/usr/lib/libnsl.so.1**

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsu
MT-Level	Safe with exceptions

**SEE ALSO**

**pvs(1), intro(2), intro(3), intro(4), attributes(5)**

<b>NAME</b>	libpam – interface library for PAM (Pluggable Authentication Module)																			
<b>SYNOPSIS</b>	cc [ <i>flag</i> ... ] <i>file</i> ... -lpam [ <i>library</i> ... ] #include <security/pam_appl.h>																			
<b>DESCRIPTION</b>	The shared object <b>libpam.so.1</b> provides the public interfaces defined below. For additional information on shared object interfaces, see <b>intro(4)</b> .																			
<b>INTERFACES</b>	<p>SUNW_1.1 (generic):</p> <table border="0"> <tr> <td><b>pam_acct_mgm</b></td> <td><b>pam_authenticate</b></td> </tr> <tr> <td><b>pam_chauthtok</b></td> <td><b>pam_close_session</b></td> </tr> <tr> <td><b>pam_end</b></td> <td><b>pam_get_data</b></td> </tr> <tr> <td><b>pam_get_item</b></td> <td><b>pam_get_user</b></td> </tr> <tr> <td><b>pam_open_session</b></td> <td><b>pam_setcred</b></td> </tr> <tr> <td><b>pam_set_data</b></td> <td><b>pam_set_item</b></td> </tr> <tr> <td><b>pam_start</b></td> <td><b>pam_strerror</b></td> </tr> </table> <p>SUNW_1.2 (generic):</p> <table border="0"> <tr> <td><b>pam_getenv</b></td> <td><b>pam_getenvlist</b></td> </tr> <tr> <td><b>pam_putenv</b></td> <td></td> </tr> </table>		<b>pam_acct_mgm</b>	<b>pam_authenticate</b>	<b>pam_chauthtok</b>	<b>pam_close_session</b>	<b>pam_end</b>	<b>pam_get_data</b>	<b>pam_get_item</b>	<b>pam_get_user</b>	<b>pam_open_session</b>	<b>pam_setcred</b>	<b>pam_set_data</b>	<b>pam_set_item</b>	<b>pam_start</b>	<b>pam_strerror</b>	<b>pam_getenv</b>	<b>pam_getenvlist</b>	<b>pam_putenv</b>	
<b>pam_acct_mgm</b>	<b>pam_authenticate</b>																			
<b>pam_chauthtok</b>	<b>pam_close_session</b>																			
<b>pam_end</b>	<b>pam_get_data</b>																			
<b>pam_get_item</b>	<b>pam_get_user</b>																			
<b>pam_open_session</b>	<b>pam_setcred</b>																			
<b>pam_set_data</b>	<b>pam_set_item</b>																			
<b>pam_start</b>	<b>pam_strerror</b>																			
<b>pam_getenv</b>	<b>pam_getenvlist</b>																			
<b>pam_putenv</b>																				
<b>FILES</b>	<ul style="list-style-type: none"> <li><b>/usr/lib/libpam.so.1</b> File that implements the PAM framework library.</li> <li><b>/etc/pam.conf</b> Configuration file.</li> <li><b>/usr/lib/security/pam_dial_auth.so.1</b> Authentication management PAM module for dialups.</li> <li><b>/usr/lib/security/pam_rhosts_auth.so.1</b> Authentication management PAM modules that use <b>ruserok()</b>.</li> <li><b>/usr/lib/security/pam_sample.so.1</b> Sample PAM module.</li> <li><b>/usr/lib/security/pam_unix.so.1</b> Authentication, account, session and password management PAM module.</li> </ul>																			
<b>ATTRIBUTES</b>	See <b>attributes(5)</b> for description of the following attributes:																			
	<table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>MT Level</td> <td>MT-Safe with exceptions</td> </tr> </tbody> </table>		ATTRIBUTE TYPE	ATTRIBUTE VALUE	MT Level	MT-Safe with exceptions														
ATTRIBUTE TYPE	ATTRIBUTE VALUE																			
MT Level	MT-Safe with exceptions																			
<b>SEE ALSO</b>	<b>pvs(1)</b> , <b>pam(3)</b> , <b>intro(4)</b> , <b>pam.conf(4)</b> , <b>attributes(5)</b> , <b>pam_dial_auth(5)</b> , <b>pam_rhosts_auth(5)</b> , <b>pam_sample(5)</b> , <b>pam_unix(5)</b>																			
<b>NOTES</b>	The interfaces in <b>libpam()</b> are MT-Safe only if each thread within the multi-threaded application uses its own PAM handle.																			

<b>NAME</b>	libposix4 – POSIX.1b Realtime Extensions library																																																
<b>SYNOPSIS</b>	<p><code>cc [ <i>flag</i> ... ] <i>file</i> ... -l<b>posix4</b> [ <i>library</i> ... ]</code></p> <p>See the man pages for the individual interfaces in section 3R for information on required headers.</p>																																																
<b>DESCRIPTION</b>	<p>Functions in this library provide most of the interfaces specified by the POSIX.1b Real-time Extension. See <b>standards(5)</b>. Specifically, this includes the interfaces defined under the Asynchronous I/O, Message Passing, Process Scheduling, Realtime Signals Extension, Semaphores, Shared Memory Objects, Synchronized I/O, and Timers options. The interfaces defined under the Memory Mapped Files, Process Memory Locking, and Range Memory Locking options are provided in <b>libc(4)</b>.</p> <p>The shared object <b>libposix4.so.1</b> provides the public interfaces defined below. For additional information on shared object interfaces, see <b>intro(4)</b>.</p>																																																
<b>INTERFACES</b>	<p>SUNW_1.1 (generic):</p> <table border="0"> <tr> <td><b>aio_cancel</b></td> <td><b>aio_error</b></td> <td><b>aio_fsync</b></td> </tr> <tr> <td><b>aio_read</b></td> <td><b>aio_return</b></td> <td><b>aio_suspend</b></td> </tr> <tr> <td><b>aio_write</b></td> <td><b>clock_getres</b></td> <td><b>clock_gettime</b></td> </tr> <tr> <td><b>clock_settime</b></td> <td><b>fdatasync</b></td> <td><b>lio_listio</b></td> </tr> <tr> <td><b>mq_close</b></td> <td><b>mq_getattr</b></td> <td><b>mq_notify</b></td> </tr> <tr> <td><b>mq_open</b></td> <td><b>mq_receive</b></td> <td><b>mq_send</b></td> </tr> <tr> <td><b>mq_setattr</b></td> <td><b>mq_unlink</b></td> <td><b>nanosleep</b></td> </tr> <tr> <td><b>sched_getparam</b></td> <td><b>sched_get_priority_max</b></td> <td><b>sched_get_priority_min</b></td> </tr> <tr> <td><b>sched_getscheduler</b></td> <td><b>sched_rr_get_interval</b></td> <td><b>sched_setparam</b></td> </tr> <tr> <td><b>sched_setscheduler</b></td> <td><b>sched_yield</b></td> <td><b>sem_close</b></td> </tr> <tr> <td><b>sem_destroy</b></td> <td><b>sem_getvalue</b></td> <td><b>sem_init</b></td> </tr> <tr> <td><b>sem_open</b></td> <td><b>sem_post</b></td> <td><b>sem_trywait</b></td> </tr> <tr> <td><b>sem_unlink</b></td> <td><b>sem_wait</b></td> <td><b>shm_open</b></td> </tr> <tr> <td><b>shm_unlink</b></td> <td><b>sigqueue</b></td> <td><b>sigtimedwait</b></td> </tr> <tr> <td><b>sigwaitinfo</b></td> <td><b>timer_create</b></td> <td><b>timer_delete</b></td> </tr> <tr> <td><b>timer_getoverrun</b></td> <td><b>timer_gettime</b></td> <td><b>timer_settime</b></td> </tr> </table>	<b>aio_cancel</b>	<b>aio_error</b>	<b>aio_fsync</b>	<b>aio_read</b>	<b>aio_return</b>	<b>aio_suspend</b>	<b>aio_write</b>	<b>clock_getres</b>	<b>clock_gettime</b>	<b>clock_settime</b>	<b>fdatasync</b>	<b>lio_listio</b>	<b>mq_close</b>	<b>mq_getattr</b>	<b>mq_notify</b>	<b>mq_open</b>	<b>mq_receive</b>	<b>mq_send</b>	<b>mq_setattr</b>	<b>mq_unlink</b>	<b>nanosleep</b>	<b>sched_getparam</b>	<b>sched_get_priority_max</b>	<b>sched_get_priority_min</b>	<b>sched_getscheduler</b>	<b>sched_rr_get_interval</b>	<b>sched_setparam</b>	<b>sched_setscheduler</b>	<b>sched_yield</b>	<b>sem_close</b>	<b>sem_destroy</b>	<b>sem_getvalue</b>	<b>sem_init</b>	<b>sem_open</b>	<b>sem_post</b>	<b>sem_trywait</b>	<b>sem_unlink</b>	<b>sem_wait</b>	<b>shm_open</b>	<b>shm_unlink</b>	<b>sigqueue</b>	<b>sigtimedwait</b>	<b>sigwaitinfo</b>	<b>timer_create</b>	<b>timer_delete</b>	<b>timer_getoverrun</b>	<b>timer_gettime</b>	<b>timer_settime</b>
<b>aio_cancel</b>	<b>aio_error</b>	<b>aio_fsync</b>																																															
<b>aio_read</b>	<b>aio_return</b>	<b>aio_suspend</b>																																															
<b>aio_write</b>	<b>clock_getres</b>	<b>clock_gettime</b>																																															
<b>clock_settime</b>	<b>fdatasync</b>	<b>lio_listio</b>																																															
<b>mq_close</b>	<b>mq_getattr</b>	<b>mq_notify</b>																																															
<b>mq_open</b>	<b>mq_receive</b>	<b>mq_send</b>																																															
<b>mq_setattr</b>	<b>mq_unlink</b>	<b>nanosleep</b>																																															
<b>sched_getparam</b>	<b>sched_get_priority_max</b>	<b>sched_get_priority_min</b>																																															
<b>sched_getscheduler</b>	<b>sched_rr_get_interval</b>	<b>sched_setparam</b>																																															
<b>sched_setscheduler</b>	<b>sched_yield</b>	<b>sem_close</b>																																															
<b>sem_destroy</b>	<b>sem_getvalue</b>	<b>sem_init</b>																																															
<b>sem_open</b>	<b>sem_post</b>	<b>sem_trywait</b>																																															
<b>sem_unlink</b>	<b>sem_wait</b>	<b>shm_open</b>																																															
<b>shm_unlink</b>	<b>sigqueue</b>	<b>sigtimedwait</b>																																															
<b>sigwaitinfo</b>	<b>timer_create</b>	<b>timer_delete</b>																																															
<b>timer_getoverrun</b>	<b>timer_gettime</b>	<b>timer_settime</b>																																															
<b>FILES</b>	<b>/usr/lib/libposix4.so.1</b> shared object																																																
<b>ATTRIBUTES</b>	See <b>attributes(5)</b> for descriptions of the following attributes:																																																
	<table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>MT-Level</td> <td>Safe</td> </tr> </tbody> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	MT-Level	Safe																																												
ATTRIBUTE TYPE	ATTRIBUTE VALUE																																																
MT-Level	Safe																																																
<b>SEE ALSO</b>	<b>pvs(1)</b> , <b>intro(4)</b> , <b>libc(4)</b> , <b>attributes(5)</b> , <b>standards(5)</b>																																																

<b>NAME</b>	libpthread – POSIX threads library																																																																				
<b>SYNOPSIS</b>	<code>cc [ <i>flag</i> ... ] <i>file</i> ... -l<b>pthread</b> [ <i>library</i> ... ]</code>																																																																				
<b>DESCRIPTION</b>	<p>Functions in this library provide the POSIX threads. See <b>standards(5)</b>. This library is implemented as a <i>filter</i> on the threads library (see <b>libthread(4)</b>).</p> <p>The shared object <b>libpthread.so.1</b> provides the public interfaces defined below.</p> <p>For additional information on shared object interfaces, see <b>intro(4)</b>.</p>																																																																				
<b>INTERFACES</b>	<p>SUNW_1.1 (generic):</p> <table border="0"> <tr> <td><b>alarm</b></td> <td><b>close</b></td> </tr> <tr> <td><b>cond_broadcast</b></td> <td><b>cond_destroy</b></td> </tr> <tr> <td><b>cond_init</b></td> <td><b>cond_signal</b></td> </tr> <tr> <td><b>cond_timedwait</b></td> <td><b>cond_wait</b></td> </tr> <tr> <td><b>creat</b></td> <td><b>fcntl</b></td> </tr> <tr> <td><b>fork</b></td> <td><b>fork1</b></td> </tr> <tr> <td><b>fsync</b></td> <td><b>_getfp</b></td> </tr> <tr> <td><b>msync</b></td> <td><b>mutex_destroy</b></td> </tr> <tr> <td><b>mutex_init</b></td> <td><b>_mutex_lock</b></td> </tr> <tr> <td><b>mutex_lock</b></td> <td><b>mutex_trylock</b></td> </tr> <tr> <td><b>mutex_unlock</b></td> <td><b>open</b></td> </tr> <tr> <td><b>pause</b></td> <td><b>pthread_atfork</b></td> </tr> <tr> <td><b>pthread_attr_destroy</b></td> <td><b>pthread_attr_getdetachstate</b></td> </tr> <tr> <td><b>pthread_attr_getinheritsched</b></td> <td><b>pthread_attr_getschedparam</b></td> </tr> <tr> <td><b>pthread_attr_getschedpolicy</b></td> <td><b>pthread_attr_getscope</b></td> </tr> <tr> <td><b>pthread_attr_getstackaddr</b></td> <td><b>pthread_attr_getstacksize</b></td> </tr> <tr> <td><b>pthread_attr_init</b></td> <td><b>pthread_attr_setdetachstate</b></td> </tr> <tr> <td><b>pthread_attr_setinheritsched</b></td> <td><b>pthread_attr_setschedparam</b></td> </tr> <tr> <td><b>pthread_attr_setschedpolicy</b></td> <td><b>pthread_attr_setscope</b></td> </tr> <tr> <td><b>pthread_attr_setstackaddr</b></td> <td><b>pthread_attr_setstacksize</b></td> </tr> <tr> <td><b>pthread_cancel</b></td> <td><b>__pthread_cleanup_pop</b></td> </tr> <tr> <td><b>__pthread_cleanup_push</b></td> <td><b>pthread_condattr_destroy</b></td> </tr> <tr> <td><b>pthread_condattr_getpshared</b></td> <td><b>pthread_condattr_init</b></td> </tr> <tr> <td><b>pthread_condattr_setpshared</b></td> <td><b>pthread_cond_broadcast</b></td> </tr> <tr> <td><b>pthread_cond_destroy</b></td> <td><b>pthread_cond_init</b></td> </tr> <tr> <td><b>pthread_cond_signal</b></td> <td><b>pthread_cond_timedwait</b></td> </tr> <tr> <td><b>pthread_cond_wait</b></td> <td><b>pthread_create</b></td> </tr> <tr> <td><b>pthread_detach</b></td> <td><b>pthread_equal</b></td> </tr> <tr> <td><b>pthread_exit</b></td> <td><b>pthread_getschedparam</b></td> </tr> <tr> <td><b>pthread_getspecific</b></td> <td><b>pthread_join</b></td> </tr> <tr> <td><b>pthread_key_create</b></td> <td><b>pthread_key_delete</b></td> </tr> <tr> <td><b>pthread_kill</b></td> <td><b>pthread_mutexattr_destroy</b></td> </tr> <tr> <td><b>pthread_mutexattr_getprioceiling</b></td> <td><b>pthread_mutexattr_getprotocol</b></td> </tr> <tr> <td><b>pthread_mutexattr_getpshared</b></td> <td><b>pthread_mutexattr_init</b></td> </tr> </table>	<b>alarm</b>	<b>close</b>	<b>cond_broadcast</b>	<b>cond_destroy</b>	<b>cond_init</b>	<b>cond_signal</b>	<b>cond_timedwait</b>	<b>cond_wait</b>	<b>creat</b>	<b>fcntl</b>	<b>fork</b>	<b>fork1</b>	<b>fsync</b>	<b>_getfp</b>	<b>msync</b>	<b>mutex_destroy</b>	<b>mutex_init</b>	<b>_mutex_lock</b>	<b>mutex_lock</b>	<b>mutex_trylock</b>	<b>mutex_unlock</b>	<b>open</b>	<b>pause</b>	<b>pthread_atfork</b>	<b>pthread_attr_destroy</b>	<b>pthread_attr_getdetachstate</b>	<b>pthread_attr_getinheritsched</b>	<b>pthread_attr_getschedparam</b>	<b>pthread_attr_getschedpolicy</b>	<b>pthread_attr_getscope</b>	<b>pthread_attr_getstackaddr</b>	<b>pthread_attr_getstacksize</b>	<b>pthread_attr_init</b>	<b>pthread_attr_setdetachstate</b>	<b>pthread_attr_setinheritsched</b>	<b>pthread_attr_setschedparam</b>	<b>pthread_attr_setschedpolicy</b>	<b>pthread_attr_setscope</b>	<b>pthread_attr_setstackaddr</b>	<b>pthread_attr_setstacksize</b>	<b>pthread_cancel</b>	<b>__pthread_cleanup_pop</b>	<b>__pthread_cleanup_push</b>	<b>pthread_condattr_destroy</b>	<b>pthread_condattr_getpshared</b>	<b>pthread_condattr_init</b>	<b>pthread_condattr_setpshared</b>	<b>pthread_cond_broadcast</b>	<b>pthread_cond_destroy</b>	<b>pthread_cond_init</b>	<b>pthread_cond_signal</b>	<b>pthread_cond_timedwait</b>	<b>pthread_cond_wait</b>	<b>pthread_create</b>	<b>pthread_detach</b>	<b>pthread_equal</b>	<b>pthread_exit</b>	<b>pthread_getschedparam</b>	<b>pthread_getspecific</b>	<b>pthread_join</b>	<b>pthread_key_create</b>	<b>pthread_key_delete</b>	<b>pthread_kill</b>	<b>pthread_mutexattr_destroy</b>	<b>pthread_mutexattr_getprioceiling</b>	<b>pthread_mutexattr_getprotocol</b>	<b>pthread_mutexattr_getpshared</b>	<b>pthread_mutexattr_init</b>
<b>alarm</b>	<b>close</b>																																																																				
<b>cond_broadcast</b>	<b>cond_destroy</b>																																																																				
<b>cond_init</b>	<b>cond_signal</b>																																																																				
<b>cond_timedwait</b>	<b>cond_wait</b>																																																																				
<b>creat</b>	<b>fcntl</b>																																																																				
<b>fork</b>	<b>fork1</b>																																																																				
<b>fsync</b>	<b>_getfp</b>																																																																				
<b>msync</b>	<b>mutex_destroy</b>																																																																				
<b>mutex_init</b>	<b>_mutex_lock</b>																																																																				
<b>mutex_lock</b>	<b>mutex_trylock</b>																																																																				
<b>mutex_unlock</b>	<b>open</b>																																																																				
<b>pause</b>	<b>pthread_atfork</b>																																																																				
<b>pthread_attr_destroy</b>	<b>pthread_attr_getdetachstate</b>																																																																				
<b>pthread_attr_getinheritsched</b>	<b>pthread_attr_getschedparam</b>																																																																				
<b>pthread_attr_getschedpolicy</b>	<b>pthread_attr_getscope</b>																																																																				
<b>pthread_attr_getstackaddr</b>	<b>pthread_attr_getstacksize</b>																																																																				
<b>pthread_attr_init</b>	<b>pthread_attr_setdetachstate</b>																																																																				
<b>pthread_attr_setinheritsched</b>	<b>pthread_attr_setschedparam</b>																																																																				
<b>pthread_attr_setschedpolicy</b>	<b>pthread_attr_setscope</b>																																																																				
<b>pthread_attr_setstackaddr</b>	<b>pthread_attr_setstacksize</b>																																																																				
<b>pthread_cancel</b>	<b>__pthread_cleanup_pop</b>																																																																				
<b>__pthread_cleanup_push</b>	<b>pthread_condattr_destroy</b>																																																																				
<b>pthread_condattr_getpshared</b>	<b>pthread_condattr_init</b>																																																																				
<b>pthread_condattr_setpshared</b>	<b>pthread_cond_broadcast</b>																																																																				
<b>pthread_cond_destroy</b>	<b>pthread_cond_init</b>																																																																				
<b>pthread_cond_signal</b>	<b>pthread_cond_timedwait</b>																																																																				
<b>pthread_cond_wait</b>	<b>pthread_create</b>																																																																				
<b>pthread_detach</b>	<b>pthread_equal</b>																																																																				
<b>pthread_exit</b>	<b>pthread_getschedparam</b>																																																																				
<b>pthread_getspecific</b>	<b>pthread_join</b>																																																																				
<b>pthread_key_create</b>	<b>pthread_key_delete</b>																																																																				
<b>pthread_kill</b>	<b>pthread_mutexattr_destroy</b>																																																																				
<b>pthread_mutexattr_getprioceiling</b>	<b>pthread_mutexattr_getprotocol</b>																																																																				
<b>pthread_mutexattr_getpshared</b>	<b>pthread_mutexattr_init</b>																																																																				

pthread_mutexattr_setprioceiling	pthread_mutexattr_setprotocol
pthread_mutexattr_setpshared	pthread_mutex_destroy
pthread_mutex_getprioceiling	pthread_mutex_init
pthread_mutex_lock	pthread_mutex_setprioceiling
pthread_mutex_trylock	pthread_mutex_unlock
pthread_once	pthread_self
pthread_setcancelstate	pthread_setcanceltype
pthread_setschedparam	pthread_setspecific
pthread_sigmask	pthread_testcancel
read	rwlock_init
rw_rdlock	rw_tryrdlock
rw_trywrlock	rw_unlock
rw_wrlock	sema_destroy
sema_init	sema_post
sema_trywait	sema_wait
setitimer	sigaction
siglongjmp	sigprocmask
sigsetjmp	sigsuspend
sigwait	sleep
tcdrain	thr_continue
thr_create	thr_exit
thr_getconcurrency	thr_getprio
thr_getspecific	thr_join
thr_keycreate	thr_kill
thr_main	thr_min_stack
thr_self	thr_setconcurrency
thr_setprio	thr_setspecific
thr_sigsetmask	thr_stksegment
thr_suspend	thr_yield
wait	waitpid
write	

**FILES** /usr/lib/libpthread.so.1 shared object

**ATTRIBUTES** See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/lib/libpthread.so.1

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsu
MT-Level	Safe

**SEE ALSO** [pvs\(1\)](#), [libpthread\(3T\)](#), [libthread\(3T\)](#), [libthread\\_db\(3T\)](#), [threads\(3T\)](#), [intro\(4\)](#), [libthread\(4\)](#), [libthread\\_db\(4\)](#), [attributes\(5\)](#), [standards\(5\)](#)

<b>NAME</b>	librac – remote asynchronous calls library					
<b>SYNOPSIS</b>	cc [ <i>flag</i> ... ] <i>file</i> ... -lrac -lnsl [ <i>library</i> ... ] #include <rpc/rpc.h> #include <rpc/rac.h>					
<b>DESCRIPTION</b>	Functions in this library provide a remote asynchronous call interface to the RPC library. The shared object <b>librac.so.1</b> provides the public interfaces defined below. For additional information on shared object interfaces, see <b>intro(4)</b> .					
<b>INTERFACES</b>	SUNW_1.1 (generic):					
	clnt_create	clnt_create_vers				
	clnt_tli_create	clnt_tp_create				
	rac_drop	rac_poll				
	rac_send	rac_senderr				
	rpcb_getmaps	rpcb_gettime				
	rpcb_set	rpcb_taddr2uaddr				
	rpcb_unset	xdrrec_create				
	xdrrec_eof	xdrrec_readbytes				
	clnt_dg_create	clnt_vc_create				
	rac_recv	rpcb_getaddr				
	rpcb_rmtcall	rpcb_uaddr2taddr				
	xdrrec_endofrecord	xdrrec_skiprecord				
<b>FILES</b>	/usr/lib/librac.so.1	shared object				
	/usr/lib/librac.a	archive library				
<b>ATTRIBUTES</b>	See <b>attributes(5)</b> for descriptions of the following attributes:					
	<table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>MT-Level</td> <td>Unsafe</td> </tr> </tbody> </table>		ATTRIBUTE TYPE	ATTRIBUTE VALUE	MT-Level	Unsafe
ATTRIBUTE TYPE	ATTRIBUTE VALUE					
MT-Level	Unsafe					
<b>SEE ALSO</b>	pvs(1), rpc_rac(3N), intro(4), attributes(5)					

<b>NAME</b>	libresolv – resolver library																																												
<b>SYNOPSIS</b>	<pre>cc [ flag ... ] file ... -lresolv -lsocket -lnsl [ library ... ] #include &lt;sys/types.h&gt; #include &lt;netinet/in.h&gt; #include &lt;arpa/nameser.h&gt; #include &lt;resolv.h&gt;</pre>																																												
<b>DESCRIPTION</b>	<p>Functions in this library provide for creating, sending, and interpreting packets to the Internet domain name servers.</p> <p>By convention, <b>libresolv.so</b> is a link to one of the shared object files for the resolver, typically the most recent one.</p> <p>For additional information on shared object interfaces, see <b>intro(4)</b>.</p>																																												
<b>Interfaces</b>	<p>The <b>resolver(3N)</b> manual page, and the system include files, describe the behavior of the functions in <b>libresolv.so.2</b>.</p> <p>The shared object <b>libresolv.so.2</b> provides the public interfaces defined below.</p> <p><b>SUNW_2.1</b> (generic):</p> <table border="0"> <tr> <td><b>_getlong</b></td> <td><b>_getshort</b></td> <td><b>_res</b></td> </tr> <tr> <td><b>__dn_skipname</b></td> <td><b>__fp_query</b></td> <td><b>__hostalias</b></td> </tr> <tr> <td><b>__p_cdname</b></td> <td><b>__p_class</b></td> <td><b>__p_query</b></td> </tr> <tr> <td><b>__p_rr</b></td> <td><b>__p_time</b></td> <td><b>__p_type</b></td> </tr> <tr> <td><b>__putlong</b></td> <td><b>dn_comp</b></td> <td><b>dn_expand</b></td> </tr> <tr> <td><b>h_errno</b></td> <td><b>res_init</b></td> <td><b>res_mkquery</b></td> </tr> <tr> <td><b>res_send</b></td> <td><b>res_search</b></td> <td><b>res_query</b></td> </tr> <tr> <td><b>res_querydomain</b></td> <td></td> <td></td> </tr> </table> <p>Programs are expected to use the aliases defined in <b>&lt;resolv.h&gt;</b> rather than calling the "___" prefixed procedures, as indicated in the following table. Use of the routines in the first column is discouraged.</p> <table border="0"> <thead> <tr> <th><b>FUNCTION REFERENCED</b></th> <th><b>ALIAS TO USE</b></th> </tr> </thead> <tbody> <tr> <td><b>__dn_skipname</b></td> <td><b>dn_skipname</b></td> </tr> <tr> <td><b>__fp_query</b></td> <td><b>fp_query</b></td> </tr> <tr> <td><b>__putlong</b></td> <td><b>putlong</b></td> </tr> <tr> <td><b>__p_cdname</b></td> <td><b>p_cdname</b></td> </tr> <tr> <td><b>__p_class</b></td> <td><b>p_class</b></td> </tr> <tr> <td><b>__p_query</b></td> <td><b>p_query</b></td> </tr> <tr> <td><b>__p_rr</b></td> <td><b>p_rr</b></td> </tr> <tr> <td><b>__p_time</b></td> <td><b>p_time</b></td> </tr> <tr> <td><b>__p_type</b></td> <td><b>p_type</b></td> </tr> </tbody> </table>	<b>_getlong</b>	<b>_getshort</b>	<b>_res</b>	<b>__dn_skipname</b>	<b>__fp_query</b>	<b>__hostalias</b>	<b>__p_cdname</b>	<b>__p_class</b>	<b>__p_query</b>	<b>__p_rr</b>	<b>__p_time</b>	<b>__p_type</b>	<b>__putlong</b>	<b>dn_comp</b>	<b>dn_expand</b>	<b>h_errno</b>	<b>res_init</b>	<b>res_mkquery</b>	<b>res_send</b>	<b>res_search</b>	<b>res_query</b>	<b>res_querydomain</b>			<b>FUNCTION REFERENCED</b>	<b>ALIAS TO USE</b>	<b>__dn_skipname</b>	<b>dn_skipname</b>	<b>__fp_query</b>	<b>fp_query</b>	<b>__putlong</b>	<b>putlong</b>	<b>__p_cdname</b>	<b>p_cdname</b>	<b>__p_class</b>	<b>p_class</b>	<b>__p_query</b>	<b>p_query</b>	<b>__p_rr</b>	<b>p_rr</b>	<b>__p_time</b>	<b>p_time</b>	<b>__p_type</b>	<b>p_type</b>
<b>_getlong</b>	<b>_getshort</b>	<b>_res</b>																																											
<b>__dn_skipname</b>	<b>__fp_query</b>	<b>__hostalias</b>																																											
<b>__p_cdname</b>	<b>__p_class</b>	<b>__p_query</b>																																											
<b>__p_rr</b>	<b>__p_time</b>	<b>__p_type</b>																																											
<b>__putlong</b>	<b>dn_comp</b>	<b>dn_expand</b>																																											
<b>h_errno</b>	<b>res_init</b>	<b>res_mkquery</b>																																											
<b>res_send</b>	<b>res_search</b>	<b>res_query</b>																																											
<b>res_querydomain</b>																																													
<b>FUNCTION REFERENCED</b>	<b>ALIAS TO USE</b>																																												
<b>__dn_skipname</b>	<b>dn_skipname</b>																																												
<b>__fp_query</b>	<b>fp_query</b>																																												
<b>__putlong</b>	<b>putlong</b>																																												
<b>__p_cdname</b>	<b>p_cdname</b>																																												
<b>__p_class</b>	<b>p_class</b>																																												
<b>__p_query</b>	<b>p_query</b>																																												
<b>__p_rr</b>	<b>p_rr</b>																																												
<b>__p_time</b>	<b>p_time</b>																																												
<b>__p_type</b>	<b>p_type</b>																																												



**libresolv.so.1** is an earlier shared library file that provides the public interfaces defined below. This file is provided for the purpose of backwards compatibility. There is no plan to fix any of its defects.

The original and complete reference documentation for these routines can only be found in earlier releases.

SUNW\_1.1 (generic):

<b>dn_comp</b>	<b>dn_expand</b>	<b>dn_skipname</b>
<b>fp_query</b>	<b>_getlong</b>	<b>_getshort</b>
<b>h_errno</b>	<b>hostalias</b>	<b>p_cdname</b>
<b>p_class</b>	<b>p_query</b>	<b>p_rr</b>
<b>p_time</b>	<b>p_type</b>	<b>putlong</b>
<b>_res</b>	<b>res_init</b>	<b>res_mkquery</b>
<b>res_query</b>	<b>res_querydomain</b>	<b>res_search</b>
<b>res_send</b>	<b>strcasecmp</b>	<b>strncasecmp</b>

**FILES**

<b>/usr/lib/libresolv.so</b>	symbolic link to most recent shared object file
<b>/usr/lib/libresolv.so.1</b>	shared object file for backward compatibility
<b>/usr/lib/libresolv.so.2</b>	shared object file

**ATTRIBUTES** See **attributes(5)** for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT-Level	Unsafe

**SEE ALSO** **pvs(1)**, **resolver(3N)**, **intro(4)**, **attributes(5)**

<b>NAME</b>	librpcsoc – obsolete RPC library												
<b>SYNOPSIS</b>	cc [ <i>flag ...</i> ] <i>file ...</i> -L/usr/ucb/lib -lrpcsoc [ <i>library ...</i> ] #include <rpc/rpc.h>												
<b>DESCRIPTION</b>	<p>Functions in this library implement socket based RPC calls (using socket calls, not TLI). Applications that require this library should link it before <b>libnsl</b>, which implements the same calls over TLI.</p> <p>This library is provided for compatibility only; new applications should not link in this library.</p> <p>The shared object <b>librpcsoc.so.1</b> provides the public interfaces defined below.</p> <p>For additional information on shared object interfaces, see <b>intro(4)</b>.</p>												
<b>INTERFACES</b>	<p>SUNW_1.1 (generic):</p> <table border="0"> <tr> <td><b>clnttcp_create</b></td> <td><b>clntudp_bufcreate</b></td> <td><b>clntudp_create</b></td> </tr> <tr> <td><b>get_myaddress</b></td> <td><b>getrpcport</b></td> <td><b>rtime</b></td> </tr> <tr> <td><b>svctcp_create</b></td> <td><b>svctcp_create</b></td> <td><b>svcudp_bufcreate</b></td> </tr> <tr> <td><b>svcudp_create</b></td> <td><b>svcudp_enablecache</b></td> <td></td> </tr> </table>	<b>clnttcp_create</b>	<b>clntudp_bufcreate</b>	<b>clntudp_create</b>	<b>get_myaddress</b>	<b>getrpcport</b>	<b>rtime</b>	<b>svctcp_create</b>	<b>svctcp_create</b>	<b>svcudp_bufcreate</b>	<b>svcudp_create</b>	<b>svcudp_enablecache</b>	
<b>clnttcp_create</b>	<b>clntudp_bufcreate</b>	<b>clntudp_create</b>											
<b>get_myaddress</b>	<b>getrpcport</b>	<b>rtime</b>											
<b>svctcp_create</b>	<b>svctcp_create</b>	<b>svcudp_bufcreate</b>											
<b>svcudp_create</b>	<b>svcudp_enablecache</b>												
<b>FILES</b>	/usr/ucb/lib/librpcsoc.so.1      shared object												
<b>ATTRIBUTES</b>	See <b>attributes(5)</b> for descriptions of the following attributes:												
	<table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>MT-Level</td> <td>Unsafe</td> </tr> </tbody> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	MT-Level	Unsafe								
ATTRIBUTE TYPE	ATTRIBUTE VALUE												
MT-Level	Unsafe												
<b>SEE ALSO</b>	<b>pvs(1)</b> , <b>rpc_soc(3N)</b> , <b>intro(4)</b> , <b>libnsl(4)</b> , <b>attributes(5)</b>												

<b>NAME</b>	librpcsvc – miscellaneous RPC services library										
<b>SYNOPSIS</b>	cc [ <i>flag</i> ... ] <i>file</i> ... -lrpcsvc [ <i>library</i> ... ] #include <rpc/rpc.h> #include <rpcsvc/rstat.h>										
<b>DESCRIPTION</b>	<p>Functions in this library provide miscellaneous RPC services. See the man pages in Section 3N for the individual functions.</p> <p>The shared object <b>librpcsvc.so.1</b> provides the public interfaces defined below. For additional information on shared object interfaces, see <b>intro(4)</b>.</p>										
<b>INTERFACES</b>	<p>SUNW_1.1 (generic):</p> <table> <tr> <td><b>havedisk</b></td> <td><b>rnusers</b></td> <td><b>rstat</b></td> </tr> <tr> <td><b>rusers</b></td> <td><b>rwall</b></td> <td><b>xdr_statstime</b></td> </tr> <tr> <td><b>xdr_statsvar</b></td> <td><b>xdr_utmpidlearr</b></td> <td></td> </tr> </table>		<b>havedisk</b>	<b>rnusers</b>	<b>rstat</b>	<b>rusers</b>	<b>rwall</b>	<b>xdr_statstime</b>	<b>xdr_statsvar</b>	<b>xdr_utmpidlearr</b>	
<b>havedisk</b>	<b>rnusers</b>	<b>rstat</b>									
<b>rusers</b>	<b>rwall</b>	<b>xdr_statstime</b>									
<b>xdr_statsvar</b>	<b>xdr_utmpidlearr</b>										
<b>FILES</b>	<table> <tr> <td><b>/usr/lib/librpcsvc.so.1</b></td> <td>shared object</td> </tr> <tr> <td><b>/usr/lib/librpcsvc.a</b></td> <td>archive library</td> </tr> </table>		<b>/usr/lib/librpcsvc.so.1</b>	shared object	<b>/usr/lib/librpcsvc.a</b>	archive library					
<b>/usr/lib/librpcsvc.so.1</b>	shared object										
<b>/usr/lib/librpcsvc.a</b>	archive library										
<b>ATTRIBUTES</b>	See <b>attributes(5)</b> for descriptions of the following attributes:										
	<table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>MT-Level</td> <td>Safe</td> </tr> </tbody> </table>		ATTRIBUTE TYPE	ATTRIBUTE VALUE	MT-Level	Safe					
ATTRIBUTE TYPE	ATTRIBUTE VALUE										
MT-Level	Safe										
<b>SEE ALSO</b>	<b>pvs(1)</b> , <b>rstat(3N)</b> , <b>intro(4)</b> , <b>attributes(5)</b>										

<b>NAME</b>	libsec – File Access Control List library						
<b>SYNOPSIS</b>	<code>cc [ <i>flag</i> ... ] <i>file</i> ... -lsec [ <i>library</i> ... ]</code> <code>#include &lt;sys/acl.h&gt;</code>						
<b>DESCRIPTION</b>	Functions in this library provide comparison and manipulation of File Access Control Lists. The shared object <b>libsec.so.1</b> provides the public interfaces defined below. For additional information on shared object interfaces, see <b>intro(4)</b> .						
<b>INTERFACES</b>	SUNW_1.1 (generic): <b>aclcheck</b> <b>aclfrommode</b> <b>aclfromtext</b> <b>aclsort</b> <b>actomode</b> <b>actotext</b>						
<b>FILES</b>	<code>/usr/lib/libsec.so.1</code> shared object <code>/usr/lib/libsec.a</code> archive library						
<b>ATTRIBUTES</b>	See <b>attributes(5)</b> for descriptions of the following attributes:						
<code>/usr/lib/libsec.so.1</code>	<table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>Availability</td> <td>SUNWcsu</td> </tr> <tr> <td>MT-Level</td> <td>Unsafe</td> </tr> </tbody> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	Availability	SUNWcsu	MT-Level	Unsafe
ATTRIBUTE TYPE	ATTRIBUTE VALUE						
Availability	SUNWcsu						
MT-Level	Unsafe						
<b>SEE ALSO</b>	<b>pvs(1)</b> , <b>intro(4)</b> , <b>attributes(5)</b>						

<b>NAME</b>	libsocket – the sockets library																																																													
<b>SYNOPSIS</b>	cc [ <i>flag</i> ... ] <i>file</i> ... -lsocket [ <i>library</i> ... ]																																																													
<b>DESCRIPTION</b>	<p>Functions in this library provide routines that provide the socket internetworking interface, primarily used with the TCP/IP protocol suite.</p> <p>The shared object <b>libsocket.so.1</b> provides the public interfaces defined below.</p> <p>For additional information on shared object interfaces, see <b>intro(4)</b>.</p>																																																													
<b>INTERFACES</b>	<p><b>SISCD_2.3 (SPARC only)</b> - The SPARC Compliance Definition, revision 2.3:</p> <table border="0"> <tr> <td><b>accept</b></td> <td><b>bind</b></td> <td><b>connect</b></td> </tr> <tr> <td><b>getpeername</b></td> <td><b>getprotobyname</b></td> <td><b>getprotobynumber</b></td> </tr> <tr> <td><b>getprotoent</b></td> <td><b>getservbyname</b></td> <td><b>getservbyport</b></td> </tr> <tr> <td><b>getsockname</b></td> <td><b>getsockopt</b></td> <td><b>inet_lnaof</b></td> </tr> <tr> <td><b>inet_makeaddr</b></td> <td><b>inet_network</b></td> <td><b>listen</b></td> </tr> <tr> <td><b>recv</b></td> <td><b>recvfrom</b></td> <td><b>recvmsg</b></td> </tr> <tr> <td><b>send</b></td> <td><b>sendmsg</b></td> <td><b>sendto</b></td> </tr> <tr> <td><b>setsockopt</b></td> <td><b>shutdown</b></td> <td><b>socket</b></td> </tr> </table> <p><b>SUNW_1.1 (generic):</b></p> <table border="0"> <tr> <td><b>bindresvport</b></td> <td><b>endnetent</b></td> <td><b>endprotoent</b></td> </tr> <tr> <td><b>endservent</b></td> <td><b>ether_aton</b></td> <td><b>ether_hostton</b></td> </tr> <tr> <td><b>ether_line</b></td> <td><b>ether_ntoa</b></td> <td><b>ether_ntohost</b></td> </tr> <tr> <td><b>fcntl</b></td> <td><b>getnetbyaddr</b></td> <td><b>getnetbyaddr_r</b></td> </tr> <tr> <td><b>getnetbyname</b></td> <td><b>getnetbyname_r</b></td> <td><b>getnetent</b></td> </tr> <tr> <td><b>getnetent_r</b></td> <td><b>getprotobyname_r</b></td> <td><b>getprotobynumber_r</b></td> </tr> <tr> <td><b>getprotoent_r</b></td> <td><b>getservbyname_r</b></td> <td><b>getservbyport_r</b></td> </tr> <tr> <td><b>getservent</b></td> <td><b>getservent_r</b></td> <td><b>htonl</b></td> </tr> <tr> <td><b>htons</b></td> <td><b>ioctl</b></td> <td><b>ntohl</b></td> </tr> <tr> <td><b>ntohs</b></td> <td><b>rcmd</b></td> <td><b>rexec</b></td> </tr> <tr> <td><b>rresvport</b></td> <td><b>ruserok</b></td> <td><b>setnetent</b></td> </tr> <tr> <td><b>setprotoent</b></td> <td><b>setservent</b></td> <td><b>socketpair</b></td> </tr> </table> <p><b>SUNW_1.1 (SPARC)</b> - This interface inherits all definitions from the generic SUNW_1.1 and the SISCD_2.3.</p> <p><b>SUNW_1.1 (i386)</b> - This interface contains all definitions from SISCD_2.3, and inherits all definitions from the generic SUNW_1.1.</p>		<b>accept</b>	<b>bind</b>	<b>connect</b>	<b>getpeername</b>	<b>getprotobyname</b>	<b>getprotobynumber</b>	<b>getprotoent</b>	<b>getservbyname</b>	<b>getservbyport</b>	<b>getsockname</b>	<b>getsockopt</b>	<b>inet_lnaof</b>	<b>inet_makeaddr</b>	<b>inet_network</b>	<b>listen</b>	<b>recv</b>	<b>recvfrom</b>	<b>recvmsg</b>	<b>send</b>	<b>sendmsg</b>	<b>sendto</b>	<b>setsockopt</b>	<b>shutdown</b>	<b>socket</b>	<b>bindresvport</b>	<b>endnetent</b>	<b>endprotoent</b>	<b>endservent</b>	<b>ether_aton</b>	<b>ether_hostton</b>	<b>ether_line</b>	<b>ether_ntoa</b>	<b>ether_ntohost</b>	<b>fcntl</b>	<b>getnetbyaddr</b>	<b>getnetbyaddr_r</b>	<b>getnetbyname</b>	<b>getnetbyname_r</b>	<b>getnetent</b>	<b>getnetent_r</b>	<b>getprotobyname_r</b>	<b>getprotobynumber_r</b>	<b>getprotoent_r</b>	<b>getservbyname_r</b>	<b>getservbyport_r</b>	<b>getservent</b>	<b>getservent_r</b>	<b>htonl</b>	<b>htons</b>	<b>ioctl</b>	<b>ntohl</b>	<b>ntohs</b>	<b>rcmd</b>	<b>rexec</b>	<b>rresvport</b>	<b>ruserok</b>	<b>setnetent</b>	<b>setprotoent</b>	<b>setservent</b>	<b>socketpair</b>
<b>accept</b>	<b>bind</b>	<b>connect</b>																																																												
<b>getpeername</b>	<b>getprotobyname</b>	<b>getprotobynumber</b>																																																												
<b>getprotoent</b>	<b>getservbyname</b>	<b>getservbyport</b>																																																												
<b>getsockname</b>	<b>getsockopt</b>	<b>inet_lnaof</b>																																																												
<b>inet_makeaddr</b>	<b>inet_network</b>	<b>listen</b>																																																												
<b>recv</b>	<b>recvfrom</b>	<b>recvmsg</b>																																																												
<b>send</b>	<b>sendmsg</b>	<b>sendto</b>																																																												
<b>setsockopt</b>	<b>shutdown</b>	<b>socket</b>																																																												
<b>bindresvport</b>	<b>endnetent</b>	<b>endprotoent</b>																																																												
<b>endservent</b>	<b>ether_aton</b>	<b>ether_hostton</b>																																																												
<b>ether_line</b>	<b>ether_ntoa</b>	<b>ether_ntohost</b>																																																												
<b>fcntl</b>	<b>getnetbyaddr</b>	<b>getnetbyaddr_r</b>																																																												
<b>getnetbyname</b>	<b>getnetbyname_r</b>	<b>getnetent</b>																																																												
<b>getnetent_r</b>	<b>getprotobyname_r</b>	<b>getprotobynumber_r</b>																																																												
<b>getprotoent_r</b>	<b>getservbyname_r</b>	<b>getservbyport_r</b>																																																												
<b>getservent</b>	<b>getservent_r</b>	<b>htonl</b>																																																												
<b>htons</b>	<b>ioctl</b>	<b>ntohl</b>																																																												
<b>ntohs</b>	<b>rcmd</b>	<b>rexec</b>																																																												
<b>rresvport</b>	<b>ruserok</b>	<b>setnetent</b>																																																												
<b>setprotoent</b>	<b>setservent</b>	<b>socketpair</b>																																																												
<b>FILES</b>	<table border="0"> <tr> <td><b>/usr/lib/libsocket.so.1</b></td> <td>shared object</td> </tr> <tr> <td><b>/usr/lib/libsocket.a</b></td> <td>archive library</td> </tr> </table>		<b>/usr/lib/libsocket.so.1</b>	shared object	<b>/usr/lib/libsocket.a</b>	archive library																																																								
<b>/usr/lib/libsocket.so.1</b>	shared object																																																													
<b>/usr/lib/libsocket.a</b>	archive library																																																													

**ATTRIBUTES**

See **attributes(5)** for descriptions of the following attributes:

**/usr/lib/libsocket.so.1**

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsu
MT-Level	Safe

**SEE ALSO**

**pvs(1), intro(2), intro(3), intro(4), attributes(5)**

<b>NAME</b>	libssagent – Sun Solstice Enterprise Agent Library															
<b>SYNOPSIS</b>	cc [ <i>flag ...</i> ] <i>file ...</i> -lssagent [ <i>library ..</i> ]															
<b>DESCRIPTION</b>	The <b>libssagent</b> is a high level API library. The <b>libssagent</b> is dependent on <b>libssasmp</b> . This library contains the starting point of the request-driven engine, that always runs in the background within the subagent. It receives SNMP requests, evaluates variables, calls the appropriate functions, and sends the correct responses.															
<b>INTERFACES</b>	Object Identifier(OID) helper functions: <table> <tr> <td><b>SSAOidCmp</b></td> <td><b>SSAOidCpy</b></td> <td><b>SSAOidDup</b></td> </tr> <tr> <td><b>SSAOidNew</b></td> <td><b>SSAOidFree</b></td> <td><b>SSAOidInit</b></td> </tr> <tr> <td><b>SSAOidString</b></td> <td><b>SSAOidStrToOid</b></td> <td><b>SSAOidZero</b></td> </tr> </table> String helper functions: <table> <tr> <td><b>SSAStringCpy</b></td> <td><b>SSAStringInit</b></td> <td><b>SSAStringToChar</b></td> </tr> <tr> <td><b>SSAStringZero</b></td> <td></td> <td></td> </tr> </table>	<b>SSAOidCmp</b>	<b>SSAOidCpy</b>	<b>SSAOidDup</b>	<b>SSAOidNew</b>	<b>SSAOidFree</b>	<b>SSAOidInit</b>	<b>SSAOidString</b>	<b>SSAOidStrToOid</b>	<b>SSAOidZero</b>	<b>SSAStringCpy</b>	<b>SSAStringInit</b>	<b>SSAStringToChar</b>	<b>SSAStringZero</b>		
<b>SSAOidCmp</b>	<b>SSAOidCpy</b>	<b>SSAOidDup</b>														
<b>SSAOidNew</b>	<b>SSAOidFree</b>	<b>SSAOidInit</b>														
<b>SSAOidString</b>	<b>SSAOidStrToOid</b>	<b>SSAOidZero</b>														
<b>SSAStringCpy</b>	<b>SSAStringInit</b>	<b>SSAStringToChar</b>														
<b>SSAStringZero</b>																
<b>ATTRIBUTES</b>	See <b>attributes(5)</b> for descriptions of the following attributes: <table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>MT-Level</td> <td>Unsafe</td> </tr> </tbody> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	MT-Level	Unsafe											
ATTRIBUTE TYPE	ATTRIBUTE VALUE															
MT-Level	Unsafe															
<b>SEE ALSO</b>	<b>libssasmp(4)</b> , <b>attributes(5)</b>															

<b>NAME</b>	libssasmp – Sun Solstice Enterprise SNMP Library						
<b>SYNOPSIS</b>	cc [ <i>flag</i> ... ] <i>file</i> ... -lssasmp [ <i>library</i> .. ]						
<b>DESCRIPTION</b>	<p>The <b>libssasmp</b> library provides low-level SNMP API functions.</p> <ul style="list-style-type: none"> <li>• ASN.1 serialization (encoding/decoding) module</li> <li>• SNMP PDU development routines</li> <li>• SNMP session module</li> <li>• Low level SNMP based API functions</li> <li>• Error-handling module</li> <li>• Trace (debugging) module</li> </ul>						
<b>INTERFACES</b>	<b>SSAAgentIsAlive</b>	<b>SSAGetTrapPort</b>	<b>SSARegSubagent</b>				
	<b>SSARegSubtree</b>	<b>SSARegSubtable</b>	<b>SSASendTrap</b>				
	<b>SSASubagentOpen</b>						
<b>ATTRIBUTES</b>	See <b>attributes(5)</b> for descriptions of the following attributes:						
	<table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>MT-Level</td> <td>Unsafe</td> </tr> </tbody> </table>			ATTRIBUTE TYPE	ATTRIBUTE VALUE	MT-Level	Unsafe
ATTRIBUTE TYPE	ATTRIBUTE VALUE						
MT-Level	Unsafe						
<b>SEE ALSO</b>	<b>libssagent(4)</b> , <b>attributes(5)</b>						



<b>NAME</b>	libsys – the system library					
<b>SYNOPSIS</b>	cc [ <i>flag</i> ... ] <i>file</i> ... -l <b>sys</b> [ <i>library</i> ... ]					
<b>DESCRIPTION</b>	<p>Functions in this library provide basic system services. This library is implemented as a <i>filter</i> on the C library (see <b>libc(4)</b>).</p> <p>The shared object <b>libsys.so.1</b> provides the public interfaces defined below.</p> <p>For additional information on shared object interfaces, see <b>intro(4)</b>.</p>					
<b>INTERFACES</b>	<p><b>SYSVABI_1.3</b> (generic) - The System V Application Binary Interface, Third Edition:</p> <table border="0"> <tr> <td style="vertical-align: top;"> <ul style="list-style-type: none"> <li><b>_access</b></li> <li><b>acct</b></li> <li><b>_altzone</b></li> <li><b>_catclose</b></li> <li><b>catgets</b></li> <li><b>_chdir</b></li> <li><b>chmod</b></li> <li><b>_chroot</b></li> <li><b>close</b></li> <li><b>_creat</b></li> <li><b>_daylight</b></li> <li><b>dup</b></li> <li><b>_execl</b></li> <li><b>execle</b></li> <li><b>_execv</b></li> <li><b>execve</b></li> <li><b>_exit</b></li> <li><b>fattach</b></li> <li><b>_fchmod</b></li> <li><b>fchown</b></li> <li><b>_fdetach</b></li> <li><b>fork</b></li> <li><b>free</b></li> <li><b>_fstatvfs</b></li> <li><b>fsync</b></li> <li><b>_getcontext</b></li> <li><b>getcwd</b></li> <li><b>_geteuid</b></li> <li><b>getgid</b></li> <li><b>_getgrnam</b></li> <li><b>getgroups</b></li> <li><b>_getmsg</b></li> <li><b>getpgid</b></li> <li><b>_getpid</b></li> </ul> </td> <td style="vertical-align: top;"> <ul style="list-style-type: none"> <li><b>access</b></li> <li><b>_alarm</b></li> <li><b>atexit</b></li> <li><b>catclose</b></li> <li><b>_catopen</b></li> <li><b>chdir</b></li> <li><b>_chown</b></li> <li><b>chroot</b></li> <li><b>_closedir</b></li> <li><b>creat</b></li> <li><b>daylight</b></li> <li><b>_environ</b></li> <li><b>execl</b></li> <li><b>_execlp</b></li> <li><b>execv</b></li> <li><b>_execvp</b></li> <li><b>exit</b></li> <li><b>_fchdir</b></li> <li><b>fchmod</b></li> <li><b>_fcntl</b></li> <li><b>fdetach</b></li> <li><b>_fpathconf</b></li> <li><b>_fstat</b></li> <li><b>fstatvfs</b></li> <li><b>_ftok</b></li> <li><b>getcontext</b></li> <li><b>_getegid</b></li> <li><b>geteuid</b></li> <li><b>_getgrgid</b></li> <li><b>getgrnam</b></li> <li><b>_getlogin</b></li> <li><b>getmsg</b></li> <li><b>_getpgrp</b></li> <li><b>getpid</b></li> </ul> </td> <td style="vertical-align: top;"> <ul style="list-style-type: none"> <li><b>_acct</b></li> <li><b>alarm</b></li> <li><b>calloc</b></li> <li><b>_catgets</b></li> <li><b>catopen</b></li> <li><b>_chmod</b></li> <li><b>chown</b></li> <li><b>_close</b></li> <li><b>closedir</b></li> <li><b>_ctype</b></li> <li><b>_dup</b></li> <li><b>environ</b></li> <li><b>_execle</b></li> <li><b>execlp</b></li> <li><b>_execve</b></li> <li><b>execvp</b></li> <li><b>_fattach</b></li> <li><b>fchdir</b></li> <li><b>_fchown</b></li> <li><b>fcntl</b></li> <li><b>_fork</b></li> <li><b>fpathconf</b></li> <li><b>fstat</b></li> <li><b>_fsync</b></li> <li><b>ftok</b></li> <li><b>_getcwd</b></li> <li><b>getegid</b></li> <li><b>_getgid</b></li> <li><b>getgrgid</b></li> <li><b>_getgroups</b></li> <li><b>getlogin</b></li> <li><b>_getpgid</b></li> <li><b>getpgrp</b></li> <li><b>_getpmsg</b></li> </ul> </td> </tr> </table>			<ul style="list-style-type: none"> <li><b>_access</b></li> <li><b>acct</b></li> <li><b>_altzone</b></li> <li><b>_catclose</b></li> <li><b>catgets</b></li> <li><b>_chdir</b></li> <li><b>chmod</b></li> <li><b>_chroot</b></li> <li><b>close</b></li> <li><b>_creat</b></li> <li><b>_daylight</b></li> <li><b>dup</b></li> <li><b>_execl</b></li> <li><b>execle</b></li> <li><b>_execv</b></li> <li><b>execve</b></li> <li><b>_exit</b></li> <li><b>fattach</b></li> <li><b>_fchmod</b></li> <li><b>fchown</b></li> <li><b>_fdetach</b></li> <li><b>fork</b></li> <li><b>free</b></li> <li><b>_fstatvfs</b></li> <li><b>fsync</b></li> <li><b>_getcontext</b></li> <li><b>getcwd</b></li> <li><b>_geteuid</b></li> <li><b>getgid</b></li> <li><b>_getgrnam</b></li> <li><b>getgroups</b></li> <li><b>_getmsg</b></li> <li><b>getpgid</b></li> <li><b>_getpid</b></li> </ul>	<ul style="list-style-type: none"> <li><b>access</b></li> <li><b>_alarm</b></li> <li><b>atexit</b></li> <li><b>catclose</b></li> <li><b>_catopen</b></li> <li><b>chdir</b></li> <li><b>_chown</b></li> <li><b>chroot</b></li> <li><b>_closedir</b></li> <li><b>creat</b></li> <li><b>daylight</b></li> <li><b>_environ</b></li> <li><b>execl</b></li> <li><b>_execlp</b></li> <li><b>execv</b></li> <li><b>_execvp</b></li> <li><b>exit</b></li> <li><b>_fchdir</b></li> <li><b>fchmod</b></li> <li><b>_fcntl</b></li> <li><b>fdetach</b></li> <li><b>_fpathconf</b></li> <li><b>_fstat</b></li> <li><b>fstatvfs</b></li> <li><b>_ftok</b></li> <li><b>getcontext</b></li> <li><b>_getegid</b></li> <li><b>geteuid</b></li> <li><b>_getgrgid</b></li> <li><b>getgrnam</b></li> <li><b>_getlogin</b></li> <li><b>getmsg</b></li> <li><b>_getpgrp</b></li> <li><b>getpid</b></li> </ul>	<ul style="list-style-type: none"> <li><b>_acct</b></li> <li><b>alarm</b></li> <li><b>calloc</b></li> <li><b>_catgets</b></li> <li><b>catopen</b></li> <li><b>_chmod</b></li> <li><b>chown</b></li> <li><b>_close</b></li> <li><b>closedir</b></li> <li><b>_ctype</b></li> <li><b>_dup</b></li> <li><b>environ</b></li> <li><b>_execle</b></li> <li><b>execlp</b></li> <li><b>_execve</b></li> <li><b>execvp</b></li> <li><b>_fattach</b></li> <li><b>fchdir</b></li> <li><b>_fchown</b></li> <li><b>fcntl</b></li> <li><b>_fork</b></li> <li><b>fpathconf</b></li> <li><b>fstat</b></li> <li><b>_fsync</b></li> <li><b>ftok</b></li> <li><b>_getcwd</b></li> <li><b>getegid</b></li> <li><b>_getgid</b></li> <li><b>getgrgid</b></li> <li><b>_getgroups</b></li> <li><b>getlogin</b></li> <li><b>_getpgid</b></li> <li><b>getpgrp</b></li> <li><b>_getpmsg</b></li> </ul>
<ul style="list-style-type: none"> <li><b>_access</b></li> <li><b>acct</b></li> <li><b>_altzone</b></li> <li><b>_catclose</b></li> <li><b>catgets</b></li> <li><b>_chdir</b></li> <li><b>chmod</b></li> <li><b>_chroot</b></li> <li><b>close</b></li> <li><b>_creat</b></li> <li><b>_daylight</b></li> <li><b>dup</b></li> <li><b>_execl</b></li> <li><b>execle</b></li> <li><b>_execv</b></li> <li><b>execve</b></li> <li><b>_exit</b></li> <li><b>fattach</b></li> <li><b>_fchmod</b></li> <li><b>fchown</b></li> <li><b>_fdetach</b></li> <li><b>fork</b></li> <li><b>free</b></li> <li><b>_fstatvfs</b></li> <li><b>fsync</b></li> <li><b>_getcontext</b></li> <li><b>getcwd</b></li> <li><b>_geteuid</b></li> <li><b>getgid</b></li> <li><b>_getgrnam</b></li> <li><b>getgroups</b></li> <li><b>_getmsg</b></li> <li><b>getpgid</b></li> <li><b>_getpid</b></li> </ul>	<ul style="list-style-type: none"> <li><b>access</b></li> <li><b>_alarm</b></li> <li><b>atexit</b></li> <li><b>catclose</b></li> <li><b>_catopen</b></li> <li><b>chdir</b></li> <li><b>_chown</b></li> <li><b>chroot</b></li> <li><b>_closedir</b></li> <li><b>creat</b></li> <li><b>daylight</b></li> <li><b>_environ</b></li> <li><b>execl</b></li> <li><b>_execlp</b></li> <li><b>execv</b></li> <li><b>_execvp</b></li> <li><b>exit</b></li> <li><b>_fchdir</b></li> <li><b>fchmod</b></li> <li><b>_fcntl</b></li> <li><b>fdetach</b></li> <li><b>_fpathconf</b></li> <li><b>_fstat</b></li> <li><b>fstatvfs</b></li> <li><b>_ftok</b></li> <li><b>getcontext</b></li> <li><b>_getegid</b></li> <li><b>geteuid</b></li> <li><b>_getgrgid</b></li> <li><b>getgrnam</b></li> <li><b>_getlogin</b></li> <li><b>getmsg</b></li> <li><b>_getpgrp</b></li> <li><b>getpid</b></li> </ul>	<ul style="list-style-type: none"> <li><b>_acct</b></li> <li><b>alarm</b></li> <li><b>calloc</b></li> <li><b>_catgets</b></li> <li><b>catopen</b></li> <li><b>_chmod</b></li> <li><b>chown</b></li> <li><b>_close</b></li> <li><b>closedir</b></li> <li><b>_ctype</b></li> <li><b>_dup</b></li> <li><b>environ</b></li> <li><b>_execle</b></li> <li><b>execlp</b></li> <li><b>_execve</b></li> <li><b>execvp</b></li> <li><b>_fattach</b></li> <li><b>fchdir</b></li> <li><b>_fchown</b></li> <li><b>fcntl</b></li> <li><b>_fork</b></li> <li><b>fpathconf</b></li> <li><b>fstat</b></li> <li><b>_fsync</b></li> <li><b>ftok</b></li> <li><b>_getcwd</b></li> <li><b>getegid</b></li> <li><b>_getgid</b></li> <li><b>getgrgid</b></li> <li><b>_getgroups</b></li> <li><b>getlogin</b></li> <li><b>_getpgid</b></li> <li><b>getpgrp</b></li> <li><b>_getpmsg</b></li> </ul>				

<b>getpmsg</b>	<b>_getppid</b>	<b>getppid</b>
<b>_getpwnam</b>	<b>getpwnam</b>	<b>_getpwuid</b>
<b>getpwuid</b>	<b>_getrlimit</b>	<b>getrlimit</b>
<b>_getsid</b>	<b>getsid</b>	<b>_gettxt</b>
<b>gettxt</b>	<b>_getuid</b>	<b>getuid</b>
<b>_grantpt</b>	<b>grantpt</b>	<b>_initgroups</b>
<b>initgroups</b>	<b>_ioctl</b>	<b>ioctl</b>
<b>_isastream</b>	<b>isastream</b>	<b>_kill</b>
<b>kill</b>	<b>_lchown</b>	<b>lchown</b>
<b>_link</b>	<b>link</b>	<b>localeconv</b>
<b>_lseek</b>	<b>lseek</b>	<b>_lstat</b>
<b>lstat</b>	<b>_makecontext</b>	<b>makecontext</b>
<b>malloc</b>	<b>_memcntl</b>	<b>memcntl</b>
<b>_mkdir</b>	<b>mkdir</b>	<b>_mknod</b>
<b>mknod</b>	<b>_mlock</b>	<b>mlock</b>
<b>_mmap</b>	<b>mmap</b>	<b>_mount</b>
<b>mount</b>	<b>_mprotect</b>	<b>mprotect</b>
<b>_msgctl</b>	<b>msgctl</b>	<b>_msgget</b>
<b>msgget</b>	<b>_msgrcv</b>	<b>msgrcv</b>
<b>_msgsnd</b>	<b>msgsnd</b>	<b>_msync</b>
<b>msync</b>	<b>_munlock</b>	<b>munlock</b>
<b>_munmap</b>	<b>munmap</b>	<b>_nice</b>
<b>nice</b>	<b>_numeric</b>	<b>_open</b>
<b>open</b>	<b>_opendir</b>	<b>opendir</b>
<b>_pathconf</b>	<b>pathconf</b>	<b>_pause</b>
<b>pause</b>	<b>_pipe</b>	<b>pipe</b>
<b>_poll</b>	<b>poll</b>	<b>_profil</b>
<b>profil</b>	<b>_ptrace</b>	<b>ptrace</b>
<b>_ptsname</b>	<b>ptsname</b>	<b>_putmsg</b>
<b>putmsg</b>	<b>_putpmsg</b>	<b>putpmsg</b>
<b>_read</b>	<b>read</b>	<b>_readdir</b>
<b>readdir</b>	<b>_readlink</b>	<b>readlink</b>
<b>_readv</b>	<b>readv</b>	<b>realloc</b>
<b>remove</b>	<b>_rename</b>	<b>rename</b>
<b>_rewinddir</b>	<b>rewinddir</b>	<b>_rmdir</b>
<b>rmdir</b>	<b>_seekdir</b>	<b>seekdir</b>
<b>_semctl</b>	<b>semctl</b>	<b>_semget</b>
<b>semget</b>	<b>_semop</b>	<b>semop</b>
<b>_setcontext</b>	<b>setcontext</b>	<b>_setgid</b>
<b>setgid</b>	<b>_setgroups</b>	<b>setgroups</b>
<b>setlocale</b>	<b>_setpgid</b>	<b>setpgid</b>
<b>_setpgrp</b>	<b>setpgrp</b>	<b>_setrlimit</b>
<b>setrlimit</b>	<b>_setsid</b>	<b>setsid</b>
<b>_setuid</b>	<b>setuid</b>	<b>_shmat</b>
<b>shmat</b>	<b>_shmctl</b>	<b>shmctl</b>

<b>_shmdt</b>	<b>shmdt</b>	<b>_shmget</b>
<b>shmget</b>	<b>_sigaction</b>	<b>sigaction</b>
<b>_sigaddset</b>	<b>sigaddset</b>	<b>_sigaltstack</b>
<b>sigaltstack</b>	<b>_sigdelset</b>	<b>sigdelset</b>
<b>_sigemptyset</b>	<b>sigemptyset</b>	<b>_sigfillset</b>
<b>sigfillset</b>	<b>_sighold</b>	<b>sighold</b>
<b>_sigignore</b>	<b>sigignore</b>	<b>_sigismember</b>
<b>sigismember</b>	<b>_siglongjmp</b>	<b>siglongjmp</b>
<b>signal</b>	<b>_sigpause</b>	<b>sigpause</b>
<b>_sigpending</b>	<b>sigpending</b>	<b>_sigprocmask</b>
<b>sigprocmask</b>	<b>_sigrelse</b>	<b>sigrelse</b>
<b>_sigsend</b>	<b>sigsend</b>	<b>_sigsendset</b>
<b>sigsendset</b>	<b>_sigset</b>	<b>sigset</b>
<b>_sigsetjmp</b>	<b>sigsetjmp</b>	<b>_sigsuspend</b>
<b>sigsuspend</b>	<b>_stat</b>	<b>stat</b>
<b>_statvfs</b>	<b>statvfs</b>	<b>_stime</b>
<b>stime</b>	<b>strcoll</b>	<b>strerror</b>
<b>strftime</b>	<b>strxfrm</b>	<b>_swapcontext</b>
<b>swapcontext</b>	<b>_symlink</b>	<b>symlink</b>
<b>_sync</b>	<b>sync</b>	<b>_sysconf</b>
<b>sysconf</b>	<b>system</b>	<b>_telldir</b>
<b>telldir</b>	<b>_time</b>	<b>time</b>
<b>_times</b>	<b>times</b>	<b>_timezone</b>
<b>timezone</b>	<b>_ttyname</b>	<b>ttyname</b>
<b>_tzname</b>	<b>tzname</b>	<b>_ulimit</b>
<b>ulimit</b>	<b>_umask</b>	<b>umask</b>
<b>_umount</b>	<b>umount</b>	<b>_uname</b>
<b>uname</b>	<b>_unlink</b>	<b>unlink</b>
<b>_unlockpt</b>	<b>unlockpt</b>	<b>_utime</b>
<b>utime</b>	<b>_wait</b>	<b>wait</b>
<b>_waitid</b>	<b>waitid</b>	<b>_waitpid</b>
<b>waitpid</b>	<b>_write</b>	<b>write</b>
<b>_writev</b>	<b>writev</b>	

**SYSVABI\_1.3 (SPARC) -**

The SPARC Processor Supplement. This interface contains all of the generic SYSVABI\_1.3, and defines:

<b>_Q_add</b>	<b>_Q_cmp</b>	<b>_Q_cmpe</b>
<b>_Q_div</b>	<b>_Q_dtoq</b>	<b>_Q_feq</b>
<b>_Q_fge</b>	<b>_Q_fgt</b>	<b>_Q_fle</b>
<b>_Qflt</b>	<b>_Q_fne</b>	<b>_Q_itoq</b>
<b>_Q_mul</b>	<b>_Q_neg</b>	<b>_Q_qtod</b>
<b>_Q_qtoi</b>	<b>_Q_qtos</b>	<b>_Q_qtou</b>
<b>_Q_sqrt</b>	<b>_Q_stoq</b>	<b>_Q_sub</b>
<b>_Q_utoq</b>	<b>.div</b>	<b>__dtou</b>
<b>_ftou</b>	<b>__huge_val</b>	<b>.mul</b>

<b>.rem</b>	<b>.stret1</b>	<b>.stret2</b>
<b>.stret4</b>	<b>.stret8</b>	<b>.udiv</b>
<b>.umul</b>	<b>.urem</b>	

**SYSVABI\_1.3 (i386)** - The Intel386 Processor Supplement. This interface contains all of the generic SYSVABI\_1.3, and defines:

<b>__flt_rounds</b>	<b>_fp_hw</b>	<b>_fpstart</b>
<b>__fxstat</b>	<b>__huge_val</b>	<b>_lxstat</b>
<b>_nuname</b>	<b>nuname</b>	<b>_sbrk</b>
<b>sbrk</b>	<b>_xmknod</b>	<b>_xstat</b>

**SISCD\_2.3 (SPARC only)** - The SPARC Compliance Definition, revision 2.3. This interface inherits all definitions from SYSVABI\_1.3.

**FILES** /usr/lib/libsys.so.1 shared object

**ATTRIBUTES** See **attributes(5)** for descriptions of the following attributes:

/usr/lib/libc.so.1

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsu
MT-Level	Safe

**SEE ALSO** **pvs(1), intro(2), intro(3), intro(4), libc(4), attributes(5)**

<b>NAME</b>	libthread – the threads library																																																																
<b>SYNOPSIS</b>	<code>cc [ <i>flag</i> ... ] <i>file</i> ... -lthread [ <i>library</i> ... ]</code>																																																																
<b>DESCRIPTION</b>	<p>Functions in this library provide routines that provide threading support. The shared object <b>libthread.so.1</b> provides the public interfaces defined below. For additional information on shared object interfaces, see <b>intro(4)</b>.</p>																																																																
<b>INTERFACES</b>	<p><b>SISCD_2.3 (SPARC only)</b> - The SPARC Compliance Definition, revision 2.3:</p> <table border="0"> <tr><td><code>cond_broadcast</code></td><td><code>cond_destroy</code></td></tr> <tr><td><code>cond_init</code></td><td><code>cond_signal</code></td></tr> <tr><td><code>cond_timedwait</code></td><td><code>fork1</code></td></tr> <tr><td><code>mutex_destroy</code></td><td><code>mutex_init</code></td></tr> <tr><td><code>mutex_lock</code></td><td><code>mutex_trylock</code></td></tr> <tr><td><code>mutex_unlock</code></td><td><code>rwlock_destroy</code></td></tr> <tr><td><code>rwlock_init</code></td><td><code>rw_rdlock</code></td></tr> <tr><td><code>rw_tryrdlock</code></td><td><code>rw_trywrlock</code></td></tr> <tr><td><code>rw_unlock</code></td><td><code>rw_wrlock</code></td></tr> <tr><td><code>sema_destroy</code></td><td><code>sema_init</code></td></tr> <tr><td><code>sema_post</code></td><td><code>sema_trywait</code></td></tr> <tr><td><code>sema_wait</code></td><td><code>sigwait</code></td></tr> <tr><td><code>thr_continue</code></td><td><code>thr_create</code></td></tr> <tr><td><code>thr_exit</code></td><td><code>thr_getconcurrency</code></td></tr> <tr><td><code>thr_getprio</code></td><td><code>thr_getspecific</code></td></tr> <tr><td><code>thr_join</code></td><td><code>thr_keycreate</code></td></tr> <tr><td><code>thr_kill</code></td><td><code>thr_main</code></td></tr> <tr><td><code>thr_min_stack</code></td><td><code>thr_self</code></td></tr> <tr><td><code>thr_setconcurrency</code></td><td><code>thr_setprio</code></td></tr> <tr><td><code>thr_setspecific</code></td><td><code>thr_sigsetmask</code></td></tr> <tr><td><code>thr_stksegment</code></td><td><code>thr_suspend</code></td></tr> <tr><td><code>thr_yield</code></td><td></td></tr> </table> <p><b>SUNW_1.1 (generic):</b></p> <table border="0"> <tr><td><code>alarm</code></td><td><code>close</code></td></tr> <tr><td><code>creat</code></td><td><code>fcntl</code></td></tr> <tr><td><code>fork</code></td><td><code>fsync</code></td></tr> <tr><td><code>_getfp</code></td><td><code>lwp_self</code></td></tr> <tr><td><code>msync</code></td><td><code>_mutex_held</code></td></tr> <tr><td><code>_mutex_lock</code></td><td><code>open</code></td></tr> <tr><td><code>pause</code></td><td><code>pthread_atfork</code></td></tr> <tr><td><code>pthread_attr_destroy</code></td><td><code>pthread_attr_getdetachstate</code></td></tr> <tr><td><code>pthread_attr_getinheritsched</code></td><td><code>pthread_attr_getschedparam</code></td></tr> <tr><td><code>pthread_attr_getschedpolicy</code></td><td><code>pthread_attr_getscope</code></td></tr> </table>	<code>cond_broadcast</code>	<code>cond_destroy</code>	<code>cond_init</code>	<code>cond_signal</code>	<code>cond_timedwait</code>	<code>fork1</code>	<code>mutex_destroy</code>	<code>mutex_init</code>	<code>mutex_lock</code>	<code>mutex_trylock</code>	<code>mutex_unlock</code>	<code>rwlock_destroy</code>	<code>rwlock_init</code>	<code>rw_rdlock</code>	<code>rw_tryrdlock</code>	<code>rw_trywrlock</code>	<code>rw_unlock</code>	<code>rw_wrlock</code>	<code>sema_destroy</code>	<code>sema_init</code>	<code>sema_post</code>	<code>sema_trywait</code>	<code>sema_wait</code>	<code>sigwait</code>	<code>thr_continue</code>	<code>thr_create</code>	<code>thr_exit</code>	<code>thr_getconcurrency</code>	<code>thr_getprio</code>	<code>thr_getspecific</code>	<code>thr_join</code>	<code>thr_keycreate</code>	<code>thr_kill</code>	<code>thr_main</code>	<code>thr_min_stack</code>	<code>thr_self</code>	<code>thr_setconcurrency</code>	<code>thr_setprio</code>	<code>thr_setspecific</code>	<code>thr_sigsetmask</code>	<code>thr_stksegment</code>	<code>thr_suspend</code>	<code>thr_yield</code>		<code>alarm</code>	<code>close</code>	<code>creat</code>	<code>fcntl</code>	<code>fork</code>	<code>fsync</code>	<code>_getfp</code>	<code>lwp_self</code>	<code>msync</code>	<code>_mutex_held</code>	<code>_mutex_lock</code>	<code>open</code>	<code>pause</code>	<code>pthread_atfork</code>	<code>pthread_attr_destroy</code>	<code>pthread_attr_getdetachstate</code>	<code>pthread_attr_getinheritsched</code>	<code>pthread_attr_getschedparam</code>	<code>pthread_attr_getschedpolicy</code>	<code>pthread_attr_getscope</code>
<code>cond_broadcast</code>	<code>cond_destroy</code>																																																																
<code>cond_init</code>	<code>cond_signal</code>																																																																
<code>cond_timedwait</code>	<code>fork1</code>																																																																
<code>mutex_destroy</code>	<code>mutex_init</code>																																																																
<code>mutex_lock</code>	<code>mutex_trylock</code>																																																																
<code>mutex_unlock</code>	<code>rwlock_destroy</code>																																																																
<code>rwlock_init</code>	<code>rw_rdlock</code>																																																																
<code>rw_tryrdlock</code>	<code>rw_trywrlock</code>																																																																
<code>rw_unlock</code>	<code>rw_wrlock</code>																																																																
<code>sema_destroy</code>	<code>sema_init</code>																																																																
<code>sema_post</code>	<code>sema_trywait</code>																																																																
<code>sema_wait</code>	<code>sigwait</code>																																																																
<code>thr_continue</code>	<code>thr_create</code>																																																																
<code>thr_exit</code>	<code>thr_getconcurrency</code>																																																																
<code>thr_getprio</code>	<code>thr_getspecific</code>																																																																
<code>thr_join</code>	<code>thr_keycreate</code>																																																																
<code>thr_kill</code>	<code>thr_main</code>																																																																
<code>thr_min_stack</code>	<code>thr_self</code>																																																																
<code>thr_setconcurrency</code>	<code>thr_setprio</code>																																																																
<code>thr_setspecific</code>	<code>thr_sigsetmask</code>																																																																
<code>thr_stksegment</code>	<code>thr_suspend</code>																																																																
<code>thr_yield</code>																																																																	
<code>alarm</code>	<code>close</code>																																																																
<code>creat</code>	<code>fcntl</code>																																																																
<code>fork</code>	<code>fsync</code>																																																																
<code>_getfp</code>	<code>lwp_self</code>																																																																
<code>msync</code>	<code>_mutex_held</code>																																																																
<code>_mutex_lock</code>	<code>open</code>																																																																
<code>pause</code>	<code>pthread_atfork</code>																																																																
<code>pthread_attr_destroy</code>	<code>pthread_attr_getdetachstate</code>																																																																
<code>pthread_attr_getinheritsched</code>	<code>pthread_attr_getschedparam</code>																																																																
<code>pthread_attr_getschedpolicy</code>	<code>pthread_attr_getscope</code>																																																																

<b>pthread_attr_getstackaddr</b>	<b>pthread_attr_getstacksize</b>
<b>pthread_attr_init</b>	<b>pthread_attr_setdetachstate</b>
<b>pthread_attr_setinheritsched</b>	<b>pthread_attr_setschedparam</b>
<b>pthread_attr_setschedpolicy</b>	<b>pthread_attr_setscope</b>
<b>pthread_attr_setstackaddr</b>	<b>pthread_attr_setstacksize</b>
<b>pthread_cancel</b>	<b>__pthread_cleanup_pop</b>
<b>__pthread_cleanup_push</b>	<b>pthread_condattr_destroy</b>
<b>pthread_condattr_getpshared</b>	<b>pthread_condattr_init</b>
<b>pthread_condattr_setpshared</b>	<b>pthread_cond_broadcast</b>
<b>pthread_cond_destroy</b>	<b>pthread_cond_init</b>
<b>pthread_cond_signal</b>	<b>pthread_cond_timedwait</b>
<b>pthread_cond_wait</b>	<b>pthread_create</b>
<b>pthread_detach</b>	<b>pthread_equal</b>
<b>pthread_exit</b>	<b>pthread_getschedparam</b>
<b>pthread_getspecific</b>	<b>pthread_join</b>
<b>pthread_key_create</b>	<b>pthread_key_delete</b>
<b>pthread_kill</b>	<b>pthread_mutexattr_destroy</b>
<b>pthread_mutexattr_getprioceiling</b>	<b>pthread_mutexattr_getprotocol</b>
<b>pthread_mutexattr_getpshared</b>	<b>pthread_mutexattr_init</b>
<b>pthread_mutexattr_setprioceiling</b>	<b>pthread_mutexattr_setprotocol</b>
<b>pthread_mutexattr_setpshared</b>	<b>pthread_mutex_destroy</b>
<b>pthread_mutex_getprioceiling</b>	<b>pthread_mutex_init</b>
<b>pthread_mutex_lock</b>	<b>pthread_mutex_setprioceiling</b>
<b>pthread_mutex_trylock</b>	<b>pthread_mutex_unlock</b>
<b>pthread_once</b>	<b>pthread_self</b>
<b>pthread_setcancelstate</b>	<b>pthread_setcanceltype</b>
<b>pthread_setschedparam</b>	<b>pthread_setspecific</b>
<b>pthread_sigmask</b>	<b>pthread_testcancel</b>
<b>read</b>	<b>_rw_read_held</b>
<b>_rw_write_held</b>	<b>_sema_held</b>
<b>setcontext</b>	<b>setitimer</b>
<b>sigaction</b>	<b>sigpending</b>
<b>sigprocmask</b>	<b>sigsuspend</b>
<b>sleep</b>	<b>tcdrain</b>
<b>wait</b>	<b>waitpid</b>
<b>write</b>	

**SUNW\_1.1 (SPARC)** - This interface inherits all definitions from the generic SUNW\_1.1 and the SISCD\_2.3, and defines:

**siglongjmp**

**sigsetjmp**

**SUNW\_1.1 (i386)** - This interface contains all definitions from SISCD\_2.3, inherits all definitions from the generic SUNW\_1.1, and defines:

**siglongjmp**

**sigsetjmp**

**FILES**

**/usr/lib/libthread.so.1** shared object

**ATTRIBUTES**

See **attributes(5)** for descriptions of the following attributes:

**/usr/lib/libthread.so.1**

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsu
MT-Level	Safe

**SEE ALSO**

**pvs(1), intro(2), libpthread(3T), libthread(3T), libthread\_db(3T), threads(3T), intro(4), libpthread(4), libthread\_db(4), attributes(5)**

<b>NAME</b>	libthread_db – threads debugging library																																													
<b>SYNOPSIS</b>	<pre>cc [ flag ... ] file ... /usr/lib/libthread_db.so.1 [ library ... ] #include &lt;proc_service.h&gt; #include &lt;thread_db.h&gt;</pre>																																													
<b>AVAILABILITY</b> /usr/lib/libthread_db.so.1	SUNWcsu																																													
<b>MT-LEVEL</b>	Safe																																													
<b>DESCRIPTION</b>	<p>Functions in this library are useful for building debuggers for multi-threaded programs. The shared object <b>libthread_db.so.1</b> provides the public interfaces defined below. For additional information on shared object interfaces, see <b>intro(4)</b>.</p>																																													
<b>INTERFACES</b>	<p><b>SUNW_1.1 (generic):</b></p> <table border="0"> <tr> <td><b>td_init</b></td> <td><b>td_log</b></td> <td><b>td_ta_delete</b></td> </tr> <tr> <td><b>td_ta_get_nthreads</b></td> <td><b>td_ta_get_ph</b></td> <td><b>td_ta_map_id2thr</b></td> </tr> <tr> <td><b>td_ta_map_lwp2thr</b></td> <td><b>td_ta_new</b></td> <td><b>td_ta_thr_iter</b></td> </tr> <tr> <td><b>td_ta_tsd_iter</b></td> <td><b>td_thr_get_info</b></td> <td><b>td_thr_getfpregs</b></td> </tr> <tr> <td><b>td_thr_getgregs</b></td> <td><b>td_thr_getxregs</b></td> <td><b>td_thr_getxregsize</b></td> </tr> <tr> <td><b>td_thr_setfpregs</b></td> <td><b>td_thr_setgregs</b></td> <td><b>td_thr_setprio</b></td> </tr> <tr> <td><b>td_thr_setsigpending</b></td> <td><b>td_thr_setxregs</b></td> <td><b>td_thr_sigsetmask</b></td> </tr> <tr> <td><b>td_thr_tsd</b></td> <td><b>td_thr_validate</b></td> <td></td> </tr> </table> <p><b>SUNW_1.2 (generic):</b></p> <table border="0"> <tr> <td><b>ta_event_addr</b></td> <td><b>td_sync_get_info</b></td> <td><b>td_sync_setstate\$</b></td> </tr> <tr> <td><b>td_sync_waiters</b></td> <td><b>td_ta_clear_event</b></td> <td><b>td_ta_enable_stats</b></td> </tr> <tr> <td><b>td_ta_event_getmsg</b></td> <td><b>td_ta_get_stats</b></td> <td><b>td_ta_map_addr2sync\$</b></td> </tr> <tr> <td><b>td_ta_reset_stats</b></td> <td><b>td_ta_set_event</b></td> <td><b>td_ta_setconcurrency</b></td> </tr> <tr> <td><b>td_ta_sync_iter</b></td> <td><b>td_thr_clear_event\$</b></td> <td><b>td_thr_dbresume</b></td> </tr> <tr> <td><b>td_thr_dbsuspend</b></td> <td><b>td_thr_event_enable\$</b></td> <td><b>td_thr_event_getmsg</b></td> </tr> <tr> <td><b>td_thr_lockowner</b></td> <td><b>td_thr_set_event</b></td> <td><b>td_thr_sleepinfo\$</b></td> </tr> </table>	<b>td_init</b>	<b>td_log</b>	<b>td_ta_delete</b>	<b>td_ta_get_nthreads</b>	<b>td_ta_get_ph</b>	<b>td_ta_map_id2thr</b>	<b>td_ta_map_lwp2thr</b>	<b>td_ta_new</b>	<b>td_ta_thr_iter</b>	<b>td_ta_tsd_iter</b>	<b>td_thr_get_info</b>	<b>td_thr_getfpregs</b>	<b>td_thr_getgregs</b>	<b>td_thr_getxregs</b>	<b>td_thr_getxregsize</b>	<b>td_thr_setfpregs</b>	<b>td_thr_setgregs</b>	<b>td_thr_setprio</b>	<b>td_thr_setsigpending</b>	<b>td_thr_setxregs</b>	<b>td_thr_sigsetmask</b>	<b>td_thr_tsd</b>	<b>td_thr_validate</b>		<b>ta_event_addr</b>	<b>td_sync_get_info</b>	<b>td_sync_setstate\$</b>	<b>td_sync_waiters</b>	<b>td_ta_clear_event</b>	<b>td_ta_enable_stats</b>	<b>td_ta_event_getmsg</b>	<b>td_ta_get_stats</b>	<b>td_ta_map_addr2sync\$</b>	<b>td_ta_reset_stats</b>	<b>td_ta_set_event</b>	<b>td_ta_setconcurrency</b>	<b>td_ta_sync_iter</b>	<b>td_thr_clear_event\$</b>	<b>td_thr_dbresume</b>	<b>td_thr_dbsuspend</b>	<b>td_thr_event_enable\$</b>	<b>td_thr_event_getmsg</b>	<b>td_thr_lockowner</b>	<b>td_thr_set_event</b>	<b>td_thr_sleepinfo\$</b>
<b>td_init</b>	<b>td_log</b>	<b>td_ta_delete</b>																																												
<b>td_ta_get_nthreads</b>	<b>td_ta_get_ph</b>	<b>td_ta_map_id2thr</b>																																												
<b>td_ta_map_lwp2thr</b>	<b>td_ta_new</b>	<b>td_ta_thr_iter</b>																																												
<b>td_ta_tsd_iter</b>	<b>td_thr_get_info</b>	<b>td_thr_getfpregs</b>																																												
<b>td_thr_getgregs</b>	<b>td_thr_getxregs</b>	<b>td_thr_getxregsize</b>																																												
<b>td_thr_setfpregs</b>	<b>td_thr_setgregs</b>	<b>td_thr_setprio</b>																																												
<b>td_thr_setsigpending</b>	<b>td_thr_setxregs</b>	<b>td_thr_sigsetmask</b>																																												
<b>td_thr_tsd</b>	<b>td_thr_validate</b>																																													
<b>ta_event_addr</b>	<b>td_sync_get_info</b>	<b>td_sync_setstate\$</b>																																												
<b>td_sync_waiters</b>	<b>td_ta_clear_event</b>	<b>td_ta_enable_stats</b>																																												
<b>td_ta_event_getmsg</b>	<b>td_ta_get_stats</b>	<b>td_ta_map_addr2sync\$</b>																																												
<b>td_ta_reset_stats</b>	<b>td_ta_set_event</b>	<b>td_ta_setconcurrency</b>																																												
<b>td_ta_sync_iter</b>	<b>td_thr_clear_event\$</b>	<b>td_thr_dbresume</b>																																												
<b>td_thr_dbsuspend</b>	<b>td_thr_event_enable\$</b>	<b>td_thr_event_getmsg</b>																																												
<b>td_thr_lockowner</b>	<b>td_thr_set_event</b>	<b>td_thr_sleepinfo\$</b>																																												
<b>FILES</b>	/usr/lib/libthread_db.so.1      shared object																																													
<b>SEE ALSO</b>	<b>pvs(1)</b> , <b>libpthread(3T)</b> , <b>libthread(3T)</b> , <b>libthread_db(3T)</b> , <b>threads(3T)</b> , <b>intro(4)</b> , <b>libthread(4)</b>																																													



<b>NAME</b>	libtnfctl – library of TNF probe control routines for use by processes and the kernel																												
<b>SYNOPSIS</b>	<code>cc [ <i>flag</i> ... ] <i>file</i> ... -ltnfctl [ <i>library</i> ... ]</code> <code>#include &lt;tnf/tnfctl.h&gt;</code>																												
<b>DESCRIPTION</b>	<p>Functions in this library provide TNF probe control routines for use by processes and the kernel.</p> <p>The shared object <b>libtnfctl.so.1</b> provides the public interfaces defined below.</p> <p>For additional information on shared object interfaces, see <b>intro(4)</b>.</p>																												
<b>INTERFACES</b>	<p>SUNW_1.1 (generic):</p> <table border="0"> <tr> <td><code>tnfctl_buffer_alloc</code></td> <td><code>tnfctl_buffer_dealloc</code></td> </tr> <tr> <td><code>tnfctl_check_libs</code></td> <td><code>tnfctl_close</code></td> </tr> <tr> <td><code>tnfctl_continue</code></td> <td><code>tnfctl_exec_open</code></td> </tr> <tr> <td><code>tnfctl_filter_list_add</code></td> <td><code>tnfctl_filter_list_delete</code></td> </tr> <tr> <td><code>tnfctl_filter_list_get</code></td> <td><code>tnfctl_filter_state_set</code></td> </tr> <tr> <td><code>tnfctl_indirect_open</code></td> <td><code>tnfctl_internal_open</code></td> </tr> <tr> <td><code>tnfctl_kernel_open</code></td> <td><code>tnfctl_pid_open</code></td> </tr> <tr> <td><code>tnfctl_probe_apply</code></td> <td><code>tnfctl_probe_apply_ids</code></td> </tr> <tr> <td><code>tnfctl_probe_connect</code></td> <td><code>tnfctl_probe_disable</code></td> </tr> <tr> <td><code>tnfctl_probe_disconnect_all</code></td> <td><code>tnfctl_probe_enable</code></td> </tr> <tr> <td><code>tnfctl_probe_state_get</code></td> <td><code>tnfctl_probe_trace</code></td> </tr> <tr> <td><code>tnfctl_probe_untrace</code></td> <td><code>tnfctl_register_funcs</code></td> </tr> <tr> <td><code>tnfctl_strerror</code></td> <td><code>tnfctl_trace_attrs_get</code></td> </tr> <tr> <td><code>tnfctl_trace_state_set</code></td> <td></td> </tr> </table>	<code>tnfctl_buffer_alloc</code>	<code>tnfctl_buffer_dealloc</code>	<code>tnfctl_check_libs</code>	<code>tnfctl_close</code>	<code>tnfctl_continue</code>	<code>tnfctl_exec_open</code>	<code>tnfctl_filter_list_add</code>	<code>tnfctl_filter_list_delete</code>	<code>tnfctl_filter_list_get</code>	<code>tnfctl_filter_state_set</code>	<code>tnfctl_indirect_open</code>	<code>tnfctl_internal_open</code>	<code>tnfctl_kernel_open</code>	<code>tnfctl_pid_open</code>	<code>tnfctl_probe_apply</code>	<code>tnfctl_probe_apply_ids</code>	<code>tnfctl_probe_connect</code>	<code>tnfctl_probe_disable</code>	<code>tnfctl_probe_disconnect_all</code>	<code>tnfctl_probe_enable</code>	<code>tnfctl_probe_state_get</code>	<code>tnfctl_probe_trace</code>	<code>tnfctl_probe_untrace</code>	<code>tnfctl_register_funcs</code>	<code>tnfctl_strerror</code>	<code>tnfctl_trace_attrs_get</code>	<code>tnfctl_trace_state_set</code>	
<code>tnfctl_buffer_alloc</code>	<code>tnfctl_buffer_dealloc</code>																												
<code>tnfctl_check_libs</code>	<code>tnfctl_close</code>																												
<code>tnfctl_continue</code>	<code>tnfctl_exec_open</code>																												
<code>tnfctl_filter_list_add</code>	<code>tnfctl_filter_list_delete</code>																												
<code>tnfctl_filter_list_get</code>	<code>tnfctl_filter_state_set</code>																												
<code>tnfctl_indirect_open</code>	<code>tnfctl_internal_open</code>																												
<code>tnfctl_kernel_open</code>	<code>tnfctl_pid_open</code>																												
<code>tnfctl_probe_apply</code>	<code>tnfctl_probe_apply_ids</code>																												
<code>tnfctl_probe_connect</code>	<code>tnfctl_probe_disable</code>																												
<code>tnfctl_probe_disconnect_all</code>	<code>tnfctl_probe_enable</code>																												
<code>tnfctl_probe_state_get</code>	<code>tnfctl_probe_trace</code>																												
<code>tnfctl_probe_untrace</code>	<code>tnfctl_register_funcs</code>																												
<code>tnfctl_strerror</code>	<code>tnfctl_trace_attrs_get</code>																												
<code>tnfctl_trace_state_set</code>																													
<b>FILES</b>	<code>/usr/lib/libtnfctl.so.1</code> shared object																												
<b>ATTRIBUTES</b>	See <b>attributes(5)</b> for descriptions of the following attributes:																												
	<table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>MT Level</td> <td>MT-Safe with exceptions</td> </tr> </tbody> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	MT Level	MT-Safe with exceptions																								
ATTRIBUTE TYPE	ATTRIBUTE VALUE																												
MT Level	MT-Safe with exceptions																												
<b>SEE ALSO</b>	<b>pvs(1)</b> , <b>libtnfctl(3X)</b> , <b>tracing(3X)</b> , <b>intro(4)</b> , <b>attributes(5)</b>																												
<b>NOTES</b>	This API is MT-Safe. Multiple threads may concurrently operate on independent <b>tnfctl</b> handles, which is the typical behavior expected. <b>libtnfctl</b> does not support multiple threads operating on the same <b>tnfctl</b> handle. If this is desired, it is the client's responsibility to implement locking to ensure that two threads that use the same <b>tnfctl</b> handle are not simultaneously present in a <b>libtnfctl</b> interface.																												

<b>NAME</b>	libucb – the UCB compatibility library																																																																		
<b>SYNOPSIS</b>	<code>cc [ <i>flag</i> ... ] <i>file</i> ... -lucb [ <i>library</i> ... ]</code>																																																																		
<b>DESCRIPTION</b>	<p>Functions in this library provide BSD semantics that were removed from the System V definition.</p> <p>The shared object <b>libucb.so.1</b> provides the public interfaces defined below.</p> <p>For additional information on shared object interfaces, see <b>intro(4)</b>.</p>																																																																		
<b>INTERFACES</b>	<p>SUNW_1.1 (generic):</p> <table border="0"> <tr> <td><b>alphasort</b></td> <td><b>bcmp</b></td> <td><b>bcopy</b></td> </tr> <tr> <td><b>bzero</b></td> <td><b>flock</b></td> <td><b>fopen</b></td> </tr> <tr> <td><b>fprintf</b></td> <td><b>freopen</b></td> <td><b>fstatfs</b></td> </tr> <tr> <td><b>ftime</b></td> <td><b>getdtablesize</b></td> <td><b>gethostid</b></td> </tr> <tr> <td><b>gethostname</b></td> <td><b>getpagesize</b></td> <td><b>getpriority</b></td> </tr> <tr> <td><b>getrusage</b></td> <td><b>gettimeofday</b></td> <td><b>getwd</b></td> </tr> <tr> <td><b>index</b></td> <td><b>killpg</b></td> <td><b>longjmp</b></td> </tr> <tr> <td><b>mctl</b></td> <td><b>nice</b></td> <td><b>nlist</b></td> </tr> <tr> <td><b>printf</b></td> <td><b>psignal</b></td> <td><b>rand</b></td> </tr> <tr> <td><b>readdir</b></td> <td><b>reboot</b></td> <td><b>re_comp</b></td> </tr> <tr> <td><b>re_exec</b></td> <td><b>rindex</b></td> <td><b>scandir</b></td> </tr> <tr> <td><b>setbuffer</b></td> <td><b>sethostname</b></td> <td><b>setjmp</b></td> </tr> <tr> <td><b>setlinebuf</b></td> <td><b>setpgrp</b></td> <td><b>setpriority</b></td> </tr> <tr> <td><b>setregid</b></td> <td><b>setreuid</b></td> <td><b>settimeofday</b></td> </tr> <tr> <td><b>sigblock</b></td> <td><b>siginterrupt</b></td> <td><b>signal</b></td> </tr> <tr> <td><b>sigpause</b></td> <td><b>sigsetmask</b></td> <td><b>sigstack</b></td> </tr> <tr> <td><b>sigvec</b></td> <td><b>sigvechandler</b></td> <td><b>sleep</b></td> </tr> <tr> <td><b>sprintf</b></td> <td><b>srand</b></td> <td><b>statfs</b></td> </tr> <tr> <td><b>sys_siglist</b></td> <td><b>times</b></td> <td><b>ualarm</b></td> </tr> <tr> <td><b>usignal</b></td> <td><b>usigpause</b></td> <td><b>usleep</b></td> </tr> <tr> <td><b>vfprintf</b></td> <td><b>vprintf</b></td> <td><b>vsprintf</b></td> </tr> <tr> <td><b>wait3</b></td> <td><b>wait4</b></td> <td></td> </tr> </table>	<b>alphasort</b>	<b>bcmp</b>	<b>bcopy</b>	<b>bzero</b>	<b>flock</b>	<b>fopen</b>	<b>fprintf</b>	<b>freopen</b>	<b>fstatfs</b>	<b>ftime</b>	<b>getdtablesize</b>	<b>gethostid</b>	<b>gethostname</b>	<b>getpagesize</b>	<b>getpriority</b>	<b>getrusage</b>	<b>gettimeofday</b>	<b>getwd</b>	<b>index</b>	<b>killpg</b>	<b>longjmp</b>	<b>mctl</b>	<b>nice</b>	<b>nlist</b>	<b>printf</b>	<b>psignal</b>	<b>rand</b>	<b>readdir</b>	<b>reboot</b>	<b>re_comp</b>	<b>re_exec</b>	<b>rindex</b>	<b>scandir</b>	<b>setbuffer</b>	<b>sethostname</b>	<b>setjmp</b>	<b>setlinebuf</b>	<b>setpgrp</b>	<b>setpriority</b>	<b>setregid</b>	<b>setreuid</b>	<b>settimeofday</b>	<b>sigblock</b>	<b>siginterrupt</b>	<b>signal</b>	<b>sigpause</b>	<b>sigsetmask</b>	<b>sigstack</b>	<b>sigvec</b>	<b>sigvechandler</b>	<b>sleep</b>	<b>sprintf</b>	<b>srand</b>	<b>statfs</b>	<b>sys_siglist</b>	<b>times</b>	<b>ualarm</b>	<b>usignal</b>	<b>usigpause</b>	<b>usleep</b>	<b>vfprintf</b>	<b>vprintf</b>	<b>vsprintf</b>	<b>wait3</b>	<b>wait4</b>	
<b>alphasort</b>	<b>bcmp</b>	<b>bcopy</b>																																																																	
<b>bzero</b>	<b>flock</b>	<b>fopen</b>																																																																	
<b>fprintf</b>	<b>freopen</b>	<b>fstatfs</b>																																																																	
<b>ftime</b>	<b>getdtablesize</b>	<b>gethostid</b>																																																																	
<b>gethostname</b>	<b>getpagesize</b>	<b>getpriority</b>																																																																	
<b>getrusage</b>	<b>gettimeofday</b>	<b>getwd</b>																																																																	
<b>index</b>	<b>killpg</b>	<b>longjmp</b>																																																																	
<b>mctl</b>	<b>nice</b>	<b>nlist</b>																																																																	
<b>printf</b>	<b>psignal</b>	<b>rand</b>																																																																	
<b>readdir</b>	<b>reboot</b>	<b>re_comp</b>																																																																	
<b>re_exec</b>	<b>rindex</b>	<b>scandir</b>																																																																	
<b>setbuffer</b>	<b>sethostname</b>	<b>setjmp</b>																																																																	
<b>setlinebuf</b>	<b>setpgrp</b>	<b>setpriority</b>																																																																	
<b>setregid</b>	<b>setreuid</b>	<b>settimeofday</b>																																																																	
<b>sigblock</b>	<b>siginterrupt</b>	<b>signal</b>																																																																	
<b>sigpause</b>	<b>sigsetmask</b>	<b>sigstack</b>																																																																	
<b>sigvec</b>	<b>sigvechandler</b>	<b>sleep</b>																																																																	
<b>sprintf</b>	<b>srand</b>	<b>statfs</b>																																																																	
<b>sys_siglist</b>	<b>times</b>	<b>ualarm</b>																																																																	
<b>usignal</b>	<b>usigpause</b>	<b>usleep</b>																																																																	
<b>vfprintf</b>	<b>vprintf</b>	<b>vsprintf</b>																																																																	
<b>wait3</b>	<b>wait4</b>																																																																		
<b>FILES</b>	<table border="0"> <tr> <td><code>/usr/ucblib/libucb.so.1</code></td> <td>shared object</td> </tr> <tr> <td><code>/usr/ucblib/libucb.a</code></td> <td>archive library</td> </tr> </table>	<code>/usr/ucblib/libucb.so.1</code>	shared object	<code>/usr/ucblib/libucb.a</code>	archive library																																																														
<code>/usr/ucblib/libucb.so.1</code>	shared object																																																																		
<code>/usr/ucblib/libucb.a</code>	archive library																																																																		
<b>ATTRIBUTES</b>	See <b>attributes(5)</b> for descriptions of the following attributes:																																																																		

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT-Level	Safe with exceptions

**SEE ALSO**

**pvs(1), intro(4), attributes(5)**

<b>NAME</b>	libvolmgt – volume management library															
<b>SYNOPSIS</b>	<code>cc [ <i>flag</i> ... ] <i>file</i> ... -lvolmgt [ <i>library</i> ... ]</code> <code>#include &lt;volmgt.h&gt;</code>															
<b>DESCRIPTION</b>	Functions in this library provide access to the volume management services. The shared object <b>libvolmgt.so.1</b> provides the public interfaces defined below. For additional information on shared object interfaces, see <b>intro(4)</b> .															
<b>INTERFACES</b>	<p>SUNW_1.1 (generic):</p> <table border="0"> <tr> <td><code>media_findname</code></td> <td><code>media_getattr</code></td> <td><code>media_getid</code></td> </tr> <tr> <td><code>media_setattr</code></td> <td><code>volmgt_check</code></td> <td><code>volmgt_inuse</code></td> </tr> <tr> <td><code>volmgt_ownspath</code></td> <td><code>volmgt_root</code></td> <td><code>volmgt_running</code></td> </tr> <tr> <td><code>volmgt_symdev</code></td> <td><code>volmgt_symname</code></td> <td></td> </tr> </table> <p>SUNW_1.2 (generic):</p> <table border="0"> <tr> <td><code>volmgt_acquire</code></td> <td><code>volmgt_release</code></td> </tr> </table> <p>SUNW_1.3 (generic):</p> <table border="0"> <tr> <td><code>volmgt_feature_enabled</code></td> </tr> </table>	<code>media_findname</code>	<code>media_getattr</code>	<code>media_getid</code>	<code>media_setattr</code>	<code>volmgt_check</code>	<code>volmgt_inuse</code>	<code>volmgt_ownspath</code>	<code>volmgt_root</code>	<code>volmgt_running</code>	<code>volmgt_symdev</code>	<code>volmgt_symname</code>		<code>volmgt_acquire</code>	<code>volmgt_release</code>	<code>volmgt_feature_enabled</code>
<code>media_findname</code>	<code>media_getattr</code>	<code>media_getid</code>														
<code>media_setattr</code>	<code>volmgt_check</code>	<code>volmgt_inuse</code>														
<code>volmgt_ownspath</code>	<code>volmgt_root</code>	<code>volmgt_running</code>														
<code>volmgt_symdev</code>	<code>volmgt_symname</code>															
<code>volmgt_acquire</code>	<code>volmgt_release</code>															
<code>volmgt_feature_enabled</code>																
<b>FILES</b>	<table border="0"> <tr> <td><code>/usr/lib/libvolmgt.so.1</code></td> <td>shared object</td> </tr> <tr> <td><code>/usr/lib/libvolmgt.a</code></td> <td>archive library</td> </tr> </table>	<code>/usr/lib/libvolmgt.so.1</code>	shared object	<code>/usr/lib/libvolmgt.a</code>	archive library											
<code>/usr/lib/libvolmgt.so.1</code>	shared object															
<code>/usr/lib/libvolmgt.a</code>	archive library															
<b>ATTRIBUTES</b>	See <b>attributes(5)</b> for descriptions of the following attributes:															
	<table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>MT-Level</td> <td>Safe with exceptions</td> </tr> </tbody> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	MT-Level	Safe with exceptions											
ATTRIBUTE TYPE	ATTRIBUTE VALUE															
MT-Level	Safe with exceptions															
<b>SEE ALSO</b>	<b>pvs(1)</b> , <b>media_findname(3X)</b> , <b>intro(4)</b> , <b>attributes(5)</b>															
<b>NOTES</b>	The MT-Level for this library of interfaces is Safe, except for <b>media_findname(3X)</b> , which is Unsafe.															

<b>NAME</b>	libw – the wide character library																																																																																	
<b>SYNOPSIS</b>	cc [ <i>flag</i> ... ] <i>file</i> ... [ <i>library</i> ... ] <b>#include</b> <wchar.h>																																																																																	
<b>DESCRIPTION</b>	<p>Historically, functions in this library provided wide character translations. This functionality now resides in <b>libc(4)</b>.</p> <p>This library is maintained to provide backward compatibility for both runtime and compilation environments. The shared object version is implemented as a filter on <b>libw.so.1</b>, and the archive version is implemented as a null archive. New application development need not reference either version of <b>libw</b>.</p> <p>The shared object <b>libw.so.1</b> provides the public interfaces defined below.</p> <p>For additional information on shared object interfaces, see <b>intro(4)</b>.</p>																																																																																	
<b>INTERFACES</b>	<p>SUNW_1.1 (generic):</p> <table border="0"> <tr> <td><b>fgetwc</b></td> <td><b>fgetws</b></td> <td><b>fputwc</b></td> </tr> <tr> <td><b>fputws</b></td> <td><b>getwc</b></td> <td><b>getwchar</b></td> </tr> <tr> <td><b>getws</b></td> <td><b>isenglish</b></td> <td><b>isideogram</b></td> </tr> <tr> <td><b>isnumber</b></td> <td><b>isphonogram</b></td> <td><b>isspecial</b></td> </tr> <tr> <td><b>iswalnum</b></td> <td><b>iswalpha</b></td> <td><b>iswcntrl</b></td> </tr> <tr> <td><b>iswctype</b></td> <td><b>iswdigit</b></td> <td><b>iswgraph</b></td> </tr> <tr> <td><b>iswlower</b></td> <td><b>iswprint</b></td> <td><b>iswpunct</b></td> </tr> <tr> <td><b>iswspace</b></td> <td><b>iswupper</b></td> <td><b>iswxdigit</b></td> </tr> <tr> <td><b>putwc</b></td> <td><b>putwchar</b></td> <td><b>putws</b></td> </tr> <tr> <td><b>strtows</b></td> <td><b>towlower</b></td> <td><b>towupper</b></td> </tr> <tr> <td><b>ungetwc</b></td> <td><b>watoll</b></td> <td><b>wcat</b></td> </tr> <tr> <td><b>wcschr</b></td> <td><b>wscmp</b></td> <td><b>wscoll</b></td> </tr> <tr> <td><b>wcscpy</b></td> <td><b>wcscspn</b></td> <td><b>wcsftime</b></td> </tr> <tr> <td><b>wcslen</b></td> <td><b>wcsncat</b></td> <td><b>wcsncmp</b></td> </tr> <tr> <td><b>wcsncpy</b></td> <td><b>wcspbrk</b></td> <td><b>wcsrchr</b></td> </tr> <tr> <td><b>wcsspn</b></td> <td><b>wctod</b></td> <td><b>wctok</b></td> </tr> <tr> <td><b>wcstol</b></td> <td><b>wctoul</b></td> <td><b>wcswcs</b></td> </tr> <tr> <td><b>wcswidth</b></td> <td><b>wcsxfrm</b></td> <td><b>wctype</b></td> </tr> <tr> <td><b>wcwidth</b></td> <td><b>wscasecmp</b></td> <td><b>wcat</b></td> </tr> <tr> <td><b>wchr</b></td> <td><b>wscmp</b></td> <td><b>wcol</b></td> </tr> <tr> <td><b>wcoll</b></td> <td><b>wscopy</b></td> <td><b>wcspn</b></td> </tr> <tr> <td><b>wsdup</b></td> <td><b>wslen</b></td> <td><b>wsncasecmp</b></td> </tr> <tr> <td><b>wsncat</b></td> <td><b>wsncmp</b></td> <td><b>wsncpy</b></td> </tr> <tr> <td><b>wspbrk</b></td> <td><b>wsprintf</b></td> <td><b>wsrchr</b></td> </tr> <tr> <td><b>wsscanf</b></td> <td><b>wssp</b></td> <td><b>wstod</b></td> </tr> <tr> <td><b>wstok</b></td> <td><b>wstol</b></td> <td><b>wstoll</b></td> </tr> <tr> <td><b>wstostr</b></td> <td><b>wsxfrm</b></td> <td></td> </tr> </table>	<b>fgetwc</b>	<b>fgetws</b>	<b>fputwc</b>	<b>fputws</b>	<b>getwc</b>	<b>getwchar</b>	<b>getws</b>	<b>isenglish</b>	<b>isideogram</b>	<b>isnumber</b>	<b>isphonogram</b>	<b>isspecial</b>	<b>iswalnum</b>	<b>iswalpha</b>	<b>iswcntrl</b>	<b>iswctype</b>	<b>iswdigit</b>	<b>iswgraph</b>	<b>iswlower</b>	<b>iswprint</b>	<b>iswpunct</b>	<b>iswspace</b>	<b>iswupper</b>	<b>iswxdigit</b>	<b>putwc</b>	<b>putwchar</b>	<b>putws</b>	<b>strtows</b>	<b>towlower</b>	<b>towupper</b>	<b>ungetwc</b>	<b>watoll</b>	<b>wcat</b>	<b>wcschr</b>	<b>wscmp</b>	<b>wscoll</b>	<b>wcscpy</b>	<b>wcscspn</b>	<b>wcsftime</b>	<b>wcslen</b>	<b>wcsncat</b>	<b>wcsncmp</b>	<b>wcsncpy</b>	<b>wcspbrk</b>	<b>wcsrchr</b>	<b>wcsspn</b>	<b>wctod</b>	<b>wctok</b>	<b>wcstol</b>	<b>wctoul</b>	<b>wcswcs</b>	<b>wcswidth</b>	<b>wcsxfrm</b>	<b>wctype</b>	<b>wcwidth</b>	<b>wscasecmp</b>	<b>wcat</b>	<b>wchr</b>	<b>wscmp</b>	<b>wcol</b>	<b>wcoll</b>	<b>wscopy</b>	<b>wcspn</b>	<b>wsdup</b>	<b>wslen</b>	<b>wsncasecmp</b>	<b>wsncat</b>	<b>wsncmp</b>	<b>wsncpy</b>	<b>wspbrk</b>	<b>wsprintf</b>	<b>wsrchr</b>	<b>wsscanf</b>	<b>wssp</b>	<b>wstod</b>	<b>wstok</b>	<b>wstol</b>	<b>wstoll</b>	<b>wstostr</b>	<b>wsxfrm</b>	
<b>fgetwc</b>	<b>fgetws</b>	<b>fputwc</b>																																																																																
<b>fputws</b>	<b>getwc</b>	<b>getwchar</b>																																																																																
<b>getws</b>	<b>isenglish</b>	<b>isideogram</b>																																																																																
<b>isnumber</b>	<b>isphonogram</b>	<b>isspecial</b>																																																																																
<b>iswalnum</b>	<b>iswalpha</b>	<b>iswcntrl</b>																																																																																
<b>iswctype</b>	<b>iswdigit</b>	<b>iswgraph</b>																																																																																
<b>iswlower</b>	<b>iswprint</b>	<b>iswpunct</b>																																																																																
<b>iswspace</b>	<b>iswupper</b>	<b>iswxdigit</b>																																																																																
<b>putwc</b>	<b>putwchar</b>	<b>putws</b>																																																																																
<b>strtows</b>	<b>towlower</b>	<b>towupper</b>																																																																																
<b>ungetwc</b>	<b>watoll</b>	<b>wcat</b>																																																																																
<b>wcschr</b>	<b>wscmp</b>	<b>wscoll</b>																																																																																
<b>wcscpy</b>	<b>wcscspn</b>	<b>wcsftime</b>																																																																																
<b>wcslen</b>	<b>wcsncat</b>	<b>wcsncmp</b>																																																																																
<b>wcsncpy</b>	<b>wcspbrk</b>	<b>wcsrchr</b>																																																																																
<b>wcsspn</b>	<b>wctod</b>	<b>wctok</b>																																																																																
<b>wcstol</b>	<b>wctoul</b>	<b>wcswcs</b>																																																																																
<b>wcswidth</b>	<b>wcsxfrm</b>	<b>wctype</b>																																																																																
<b>wcwidth</b>	<b>wscasecmp</b>	<b>wcat</b>																																																																																
<b>wchr</b>	<b>wscmp</b>	<b>wcol</b>																																																																																
<b>wcoll</b>	<b>wscopy</b>	<b>wcspn</b>																																																																																
<b>wsdup</b>	<b>wslen</b>	<b>wsncasecmp</b>																																																																																
<b>wsncat</b>	<b>wsncmp</b>	<b>wsncpy</b>																																																																																
<b>wspbrk</b>	<b>wsprintf</b>	<b>wsrchr</b>																																																																																
<b>wsscanf</b>	<b>wssp</b>	<b>wstod</b>																																																																																
<b>wstok</b>	<b>wstol</b>	<b>wstoll</b>																																																																																
<b>wstostr</b>	<b>wsxfrm</b>																																																																																	

**FILES** /usr/lib/libw.so.1 a filter on **libc.so.1**  
 /usr/lib/libw.a a link to /usr/lib/null.a

**ATTRIBUTES** See **attributes(5)** for descriptions of the following attributes:

/usr/lib/libw.so.1

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsu
MT-Level	Safe

**SEE ALSO** **pvs(1)**, **intro(3)**, **intro(4)**, **libc(4)**, **attributes(5)**

<b>NAME</b>	libxfn – the XFN interface library		
<b>SYNOPSIS</b>	<pre>cc [ flag ... ] file ... -lxfn [ library ... ] #include      &lt;xfn/xfn.h&gt;</pre>		
<b>DESCRIPTION</b>	<p>This library provides the implementation of XFN, the X/Open Federated Naming specification (see <b>xfn(3N)</b> and <b>fns(5)</b>).</p> <p>The shared object <b>libxfn.so.1</b> provides the public interfaces defined below.</p> <p>For additional information on shared object interfaces, see <b>intro(4)</b>.</p>		
<b>INTERFACES</b>	<p><b>SUNW_1.1 (generic):</b></p> <table border="0" style="width: 100%;"> <tr> <td style="vertical-align: top; padding-right: 20px;"> <b>fn_attr_get</b>  <b>fn_attr_get_values</b>  <b>fn_attribute_assign</b>  <b>fn_attribute_create</b>  <b>fn_attribute_first</b>  <b>fn_attribute_next</b>  <b>fn_attribute_syntax</b>  <b>fn_attr_modify</b>  <b>fn_attrmodlist_assign</b>  <b>fn_attrmodlist_count</b>  <b>fn_attrmodlist_destroy</b>  <b>fn_attrmodlist_next</b>  <b>fn_attr_multi_modify</b>  <b>fn_attrset_assign</b>  <b>fn_attrset_count</b>  <b>fn_attrset_destroy</b>  <b>fn_attrset_get</b>  <b>fn_attrset_remove</b>  <b>fn_bindinglist_next</b>  <b>fn_bindingset_assign</b>  <b>fn_bindingset_count</b>  <b>fn_bindingset_destroy</b>  <b>fn_bindingset_get_ref</b>  <b>fn_bindingset_remove</b>  <b>fn_composite_name_append_name</b>  <b>fn_composite_name_assign_string</b>  <b>fn_composite_name_count</b>  <b>fn_composite_name_delete_comp</b>  <b>fn_composite_name_first</b>  <b>fn_composite_name_from_string</b>  <b>fn_composite_name_insert_name</b>  <b>fn_composite_name_is_equal</b> </td> <td style="vertical-align: top;"> <b>fn_attr_get_ids</b>  <b>fn_attribute_add</b>  <b>fn_attribute_copy</b>  <b>fn_attribute_destroy</b>  <b>fn_attribute_identifier</b>  <b>fn_attribute_remove</b>  <b>fn_attribute_valuecount</b>  <b>fn_attrmodlist_add</b>  <b>fn_attrmodlist_copy</b>  <b>fn_attrmodlist_create</b>  <b>fn_attrmodlist_first</b>  <b>fn_attr_multi_get</b>  <b>fn_attrset_add</b>  <b>fn_attrset_copy</b>  <b>fn_attrset_create</b>  <b>fn_attrset_first</b>  <b>fn_attrset_next</b>  <b>fn_bindinglist_destroy</b>  <b>fn_bindingset_add</b>  <b>fn_bindingset_copy</b>  <b>fn_bindingset_create</b>  <b>fn_bindingset_first</b>  <b>fn_bindingset_next</b>  <b>fn_composite_name_append_comp</b>  <b>fn_composite_name_assign</b>  <b>fn_composite_name_copy</b>  <b>fn_composite_name_create</b>  <b>fn_composite_name_destroy</b>  <b>fn_composite_name_from_str</b>  <b>fn_composite_name_insert_comp</b>  <b>fn_composite_name_is_empty</b>  <b>fn_composite_name_is_prefix</b> </td> </tr> </table>	<b>fn_attr_get</b> <b>fn_attr_get_values</b> <b>fn_attribute_assign</b> <b>fn_attribute_create</b> <b>fn_attribute_first</b> <b>fn_attribute_next</b> <b>fn_attribute_syntax</b> <b>fn_attr_modify</b> <b>fn_attrmodlist_assign</b> <b>fn_attrmodlist_count</b> <b>fn_attrmodlist_destroy</b> <b>fn_attrmodlist_next</b> <b>fn_attr_multi_modify</b> <b>fn_attrset_assign</b> <b>fn_attrset_count</b> <b>fn_attrset_destroy</b> <b>fn_attrset_get</b> <b>fn_attrset_remove</b> <b>fn_bindinglist_next</b> <b>fn_bindingset_assign</b> <b>fn_bindingset_count</b> <b>fn_bindingset_destroy</b> <b>fn_bindingset_get_ref</b> <b>fn_bindingset_remove</b> <b>fn_composite_name_append_name</b> <b>fn_composite_name_assign_string</b> <b>fn_composite_name_count</b> <b>fn_composite_name_delete_comp</b> <b>fn_composite_name_first</b> <b>fn_composite_name_from_string</b> <b>fn_composite_name_insert_name</b> <b>fn_composite_name_is_equal</b>	<b>fn_attr_get_ids</b> <b>fn_attribute_add</b> <b>fn_attribute_copy</b> <b>fn_attribute_destroy</b> <b>fn_attribute_identifier</b> <b>fn_attribute_remove</b> <b>fn_attribute_valuecount</b> <b>fn_attrmodlist_add</b> <b>fn_attrmodlist_copy</b> <b>fn_attrmodlist_create</b> <b>fn_attrmodlist_first</b> <b>fn_attr_multi_get</b> <b>fn_attrset_add</b> <b>fn_attrset_copy</b> <b>fn_attrset_create</b> <b>fn_attrset_first</b> <b>fn_attrset_next</b> <b>fn_bindinglist_destroy</b> <b>fn_bindingset_add</b> <b>fn_bindingset_copy</b> <b>fn_bindingset_create</b> <b>fn_bindingset_first</b> <b>fn_bindingset_next</b> <b>fn_composite_name_append_comp</b> <b>fn_composite_name_assign</b> <b>fn_composite_name_copy</b> <b>fn_composite_name_create</b> <b>fn_composite_name_destroy</b> <b>fn_composite_name_from_str</b> <b>fn_composite_name_insert_comp</b> <b>fn_composite_name_is_empty</b> <b>fn_composite_name_is_prefix</b>
<b>fn_attr_get</b> <b>fn_attr_get_values</b> <b>fn_attribute_assign</b> <b>fn_attribute_create</b> <b>fn_attribute_first</b> <b>fn_attribute_next</b> <b>fn_attribute_syntax</b> <b>fn_attr_modify</b> <b>fn_attrmodlist_assign</b> <b>fn_attrmodlist_count</b> <b>fn_attrmodlist_destroy</b> <b>fn_attrmodlist_next</b> <b>fn_attr_multi_modify</b> <b>fn_attrset_assign</b> <b>fn_attrset_count</b> <b>fn_attrset_destroy</b> <b>fn_attrset_get</b> <b>fn_attrset_remove</b> <b>fn_bindinglist_next</b> <b>fn_bindingset_assign</b> <b>fn_bindingset_count</b> <b>fn_bindingset_destroy</b> <b>fn_bindingset_get_ref</b> <b>fn_bindingset_remove</b> <b>fn_composite_name_append_name</b> <b>fn_composite_name_assign_string</b> <b>fn_composite_name_count</b> <b>fn_composite_name_delete_comp</b> <b>fn_composite_name_first</b> <b>fn_composite_name_from_string</b> <b>fn_composite_name_insert_name</b> <b>fn_composite_name_is_equal</b>	<b>fn_attr_get_ids</b> <b>fn_attribute_add</b> <b>fn_attribute_copy</b> <b>fn_attribute_destroy</b> <b>fn_attribute_identifier</b> <b>fn_attribute_remove</b> <b>fn_attribute_valuecount</b> <b>fn_attrmodlist_add</b> <b>fn_attrmodlist_copy</b> <b>fn_attrmodlist_create</b> <b>fn_attrmodlist_first</b> <b>fn_attr_multi_get</b> <b>fn_attrset_add</b> <b>fn_attrset_copy</b> <b>fn_attrset_create</b> <b>fn_attrset_first</b> <b>fn_attrset_next</b> <b>fn_bindinglist_destroy</b> <b>fn_bindingset_add</b> <b>fn_bindingset_copy</b> <b>fn_bindingset_create</b> <b>fn_bindingset_first</b> <b>fn_bindingset_next</b> <b>fn_composite_name_append_comp</b> <b>fn_composite_name_assign</b> <b>fn_composite_name_copy</b> <b>fn_composite_name_create</b> <b>fn_composite_name_destroy</b> <b>fn_composite_name_from_str</b> <b>fn_composite_name_insert_comp</b> <b>fn_composite_name_is_empty</b> <b>fn_composite_name_is_prefix</b>		

<code>fn_composite_name_is_suffix</code>	<code>fn_composite_name_last</code>
<code>fn_composite_name_next</code>	<code>fn_composite_name_prefix</code>
<code>fn_composite_name_prepend_comp</code>	<code>fn_composite_name_prepend_name</code>
<code>fn_composite_name_prev</code>	<code>fn_composite_name_suffix</code>
<code>fn_compound_name_append_comp</code>	<code>fn_compound_name_assign</code>
<code>fn_compound_name_copy</code>	<code>fn_compound_name_count</code>
<code>fn_compound_name_delete_all</code>	<code>fn_compound_name_delete_comp</code>
<code>fn_compound_name_destroy</code>	<code>fn_compound_name_first</code>
<code>fn_compound_name_from_syntax_attrs</code>	<code>fn_compound_name_get_syntax_attrs</code>
<code>fn_compound_name_insert_comp</code>	<code>fn_compound_name_is_empty</code>
<code>fn_compound_name_is_equal</code>	<code>fn_compound_name_is_prefix</code>
<code>fn_compound_name_is_suffix</code>	<code>fn_compound_name_last</code>
<code>fn_compound_name_next</code>	<code>fn_compound_name_prefix</code>
<code>fn_compound_name_prepend_comp</code>	<code>fn_compound_name_prev</code>
<code>fn_compound_name_suffix</code>	<code>fn_ctx_bind</code>
<code>fn_ctx_create_subcontext</code>	<code>fn_ctx_destroy_subcontext</code>
<code>fn_ctx_get_ref</code>	<code>fn_ctx_get_syntax_attrs</code>
<code>fn_ctx_handle_destroy</code>	<code>fn_ctx_handle_from_initial</code>
<code>fn_ctx_handle_from_ref</code>	<code>fn_ctx_list_bindings</code>
<code>fn_ctx_list_names</code>	<code>fn_ctx_lookup</code>
<code>fn_ctx_lookup_link</code>	<code>fn_ctx_rename</code>
<code>fn_ctx_unbind</code>	<code>fn_multigetlist_destroy</code>
<code>fn_multigetlist_next</code>	<code>fn_namelist_destroy</code>
<code>fn_namelist_next</code>	<code>fn_nameset_add</code>
<code>fn_nameset_assign</code>	<code>fn_nameset_copy</code>
<code>fn_nameset_count</code>	<code>fn_nameset_create</code>
<code>fn_nameset_destroy</code>	<code>fn_nameset_first</code>
<code>fn_nameset_next</code>	<code>fn_nameset_remove</code>
<code>fn_ref_addr_assign</code>	<code>fn_ref_addr_copy</code>
<code>fn_ref_addrcount</code>	<code>fn_ref_addr_create</code>
<code>fn_ref_addr_data</code>	<code>fn_ref_addr_description</code>
<code>fn_ref_addr_destroy</code>	<code>fn_ref_addr_length</code>
<code>fn_ref_addr_type</code>	<code>fn_ref_append_addr</code>
<code>fn_ref_assign</code>	<code>fn_ref_copy</code>
<code>fn_ref_create</code>	<code>fn_ref_create_link</code>
<code>fn_ref_delete_addr</code>	<code>fn_ref_delete_all</code>
<code>fn_ref_description</code>	<code>fn_ref_destroy</code>
<code>fn_ref_first</code>	<code>fn_ref_insert_addr</code>
<code>fn_ref_is_link</code>	<code>fn_ref_link_name</code>
<code>fn_ref_next</code>	<code>fn_ref_prepend_addr</code>
<code>fn_ref_type</code>	<code>fn_status_advance_by_name</code>
<code>fn_status_append_remaining_name</code>	<code>fn_status_append_resolved_name</code>
<code>fn_status_assign</code>	<code>fn_status_code</code>
<code>fn_status_copy</code>	<code>fn_status_create</code>
<code>fn_status_description</code>	<code>fn_status_destroy</code>



<code>fn_status_diagnostic_message</code>	<code>fn_status_is_success</code>
<code>fn_status_link_code</code>	<code>fn_status_link_diagnostic_message</code>
<code>fn_status_link_remaining_name</code>	<code>fn_status_link_resolved_name</code>
<code>fn_status_link_resolved_ref</code>	<code>fn_status_remaining_name</code>
<code>fn_status_resolved_name</code>	<code>fn_status_resolved_ref</code>
<code>fn_status_set</code>	<code>fn_status_set_code</code>
<code>fn_status_set_diagnostic_message</code>	<code>fn_status_set_link_code</code>
<code>fn_status_set_link_diagnostic_message</code>	<code>fn_status_set_link_remaining_name</code>
<code>fn_status_set_link_resolved_name</code>	<code>fn_status_set_link_resolved_ref</code>
<code>fn_status_set_remaining_name</code>	<code>fn_status_set_resolved_name</code>
<code>fn_status_set_resolved_ref</code>	<code>fn_status_set_success</code>
<code>fn_string_assign</code>	<code>fn_string_bytecount</code>
<code>fn_string_charcount</code>	<code>fn_string_code_set</code>
<code>fn_string_compare</code>	<code>fn_string_compare_substring</code>
<code>fn_string_contents</code>	<code>fn_string_copy</code>
<code>fn_string_create</code>	<code>fn_string_destroy</code>
<code>fn_string_from_composite_name</code>	<code>fn_string_from_compound_name</code>
<code>fn_string_from_contents</code>	<code>fn_string_from_str</code>
<code>fn_string_from_strings</code>	<code>fn_string_from_str_n</code>
<code>fn_string_from_substring</code>	<code>fn_string_is_empty</code>
<code>fn_string_next_substring</code>	<code>fn_string_prev_substring</code>
<code>fn_string_str</code>	<code>fn_valuelist_destroy</code>
<code>fn_valuelist_next</code>	

**FILES** /usr/lib/libxfn.so.1 shared object

**ATTRIBUTES** See `attributes(5)` for descriptions of the following attributes:

/usr/lib/libxfn.so.1

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWfns
MT-Level	Safe

**SEE ALSO** `pvs(1)`, `intro(3)`, `xfn(3N)`, `intro(4)`, `attributes(5)`, `fns(5)`

<b>NAME</b>	libxnet – X/Open Networking Interfaces library																																																																															
<b>SYNOPSIS</b>	cc [ <i>flag</i> ... ] <i>file</i> ... -lxnet [ <i>library</i> ... ]																																																																															
<b>DESCRIPTION</b>	<p>Functions in this library provide networking interfaces which comply with the X/Open CAE Specification, Networking Services, Issue 4.</p> <p>The shared object <b>libxnet.so.1</b> and its dependants provide the public interfaces defined below.</p> <p>For additional information on shared object interfaces, see <b>intro(4)</b>.</p>																																																																															
<b>INTERFACES</b>	<p>SUNW_1.1 (generic):</p> <table border="0"> <tr> <td><b>accept</b></td> <td><b>bind</b></td> <td><b>connect</b></td> </tr> <tr> <td><b>endhostent</b></td> <td><b>endnetent</b></td> <td><b>endprotoent</b></td> </tr> <tr> <td><b>endservent</b></td> <td><b>gethostbyaddr</b></td> <td><b>gethostbyname</b></td> </tr> <tr> <td><b>gethostent</b></td> <td><b>gethostbyname</b></td> <td><b>getnetbyaddr</b></td> </tr> <tr> <td><b>getnetbyname</b></td> <td><b>getnetent</b></td> <td><b>getpeername</b></td> </tr> <tr> <td><b>getprotobyname</b></td> <td><b>getprotobynumber</b></td> <td><b>getprotoent</b></td> </tr> <tr> <td><b>getservbyname</b></td> <td><b>getservbyport</b></td> <td><b>getservent</b></td> </tr> <tr> <td><b>getsockname</b></td> <td><b>getsockopt</b></td> <td><b>h_errno</b></td> </tr> <tr> <td><b>htonl</b></td> <td><b>htons</b></td> <td><b>inet_addr</b></td> </tr> <tr> <td><b>inet_lnaof</b></td> <td><b>inet_makeaddr</b></td> <td><b>inet_netof</b></td> </tr> <tr> <td><b>inet_network</b></td> <td><b>inet_ntoa</b></td> <td><b>listen</b></td> </tr> <tr> <td><b>ntohl</b></td> <td><b>ntohs</b></td> <td><b>recv</b></td> </tr> <tr> <td><b>recvfrom</b></td> <td><b>recvmsg</b></td> <td><b>send</b></td> </tr> <tr> <td><b>sendmsg</b></td> <td><b>sendto</b></td> <td><b>sethostent</b></td> </tr> <tr> <td><b>setnetent</b></td> <td><b>setprotoent</b></td> <td><b>setservent</b></td> </tr> <tr> <td><b>setsockopt</b></td> <td><b>shutdown</b></td> <td><b>socket</b></td> </tr> <tr> <td><b>socketpair</b></td> <td><b>t_accept</b></td> <td><b>t_alloc</b></td> </tr> <tr> <td><b>t_bind</b></td> <td><b>t_close</b></td> <td><b>t_connect</b></td> </tr> <tr> <td><b>t_errno</b></td> <td><b>t_error</b></td> <td><b>t_free</b></td> </tr> <tr> <td><b>t_getinfo</b></td> <td><b>t_getprotaddr</b></td> <td><b>t_getstate</b></td> </tr> <tr> <td><b>t_listen</b></td> <td><b>t_look</b></td> <td><b>t_open</b></td> </tr> <tr> <td><b>t_optmgmt</b></td> <td><b>t_rcv</b></td> <td><b>t_rcvconnect</b></td> </tr> <tr> <td><b>t_rcvdis</b></td> <td><b>t_rcvrel</b></td> <td><b>t_rcvudata</b></td> </tr> <tr> <td><b>t_rcvuderr</b></td> <td><b>t_snd</b></td> <td><b>t_snddis</b></td> </tr> <tr> <td><b>t_sndrel</b></td> <td><b>t_sndudata</b></td> <td><b>t_strerror</b></td> </tr> <tr> <td><b>t_sync</b></td> <td><b>t_unbind</b></td> <td></td> </tr> </table>		<b>accept</b>	<b>bind</b>	<b>connect</b>	<b>endhostent</b>	<b>endnetent</b>	<b>endprotoent</b>	<b>endservent</b>	<b>gethostbyaddr</b>	<b>gethostbyname</b>	<b>gethostent</b>	<b>gethostbyname</b>	<b>getnetbyaddr</b>	<b>getnetbyname</b>	<b>getnetent</b>	<b>getpeername</b>	<b>getprotobyname</b>	<b>getprotobynumber</b>	<b>getprotoent</b>	<b>getservbyname</b>	<b>getservbyport</b>	<b>getservent</b>	<b>getsockname</b>	<b>getsockopt</b>	<b>h_errno</b>	<b>htonl</b>	<b>htons</b>	<b>inet_addr</b>	<b>inet_lnaof</b>	<b>inet_makeaddr</b>	<b>inet_netof</b>	<b>inet_network</b>	<b>inet_ntoa</b>	<b>listen</b>	<b>ntohl</b>	<b>ntohs</b>	<b>recv</b>	<b>recvfrom</b>	<b>recvmsg</b>	<b>send</b>	<b>sendmsg</b>	<b>sendto</b>	<b>sethostent</b>	<b>setnetent</b>	<b>setprotoent</b>	<b>setservent</b>	<b>setsockopt</b>	<b>shutdown</b>	<b>socket</b>	<b>socketpair</b>	<b>t_accept</b>	<b>t_alloc</b>	<b>t_bind</b>	<b>t_close</b>	<b>t_connect</b>	<b>t_errno</b>	<b>t_error</b>	<b>t_free</b>	<b>t_getinfo</b>	<b>t_getprotaddr</b>	<b>t_getstate</b>	<b>t_listen</b>	<b>t_look</b>	<b>t_open</b>	<b>t_optmgmt</b>	<b>t_rcv</b>	<b>t_rcvconnect</b>	<b>t_rcvdis</b>	<b>t_rcvrel</b>	<b>t_rcvudata</b>	<b>t_rcvuderr</b>	<b>t_snd</b>	<b>t_snddis</b>	<b>t_sndrel</b>	<b>t_sndudata</b>	<b>t_strerror</b>	<b>t_sync</b>	<b>t_unbind</b>	
<b>accept</b>	<b>bind</b>	<b>connect</b>																																																																														
<b>endhostent</b>	<b>endnetent</b>	<b>endprotoent</b>																																																																														
<b>endservent</b>	<b>gethostbyaddr</b>	<b>gethostbyname</b>																																																																														
<b>gethostent</b>	<b>gethostbyname</b>	<b>getnetbyaddr</b>																																																																														
<b>getnetbyname</b>	<b>getnetent</b>	<b>getpeername</b>																																																																														
<b>getprotobyname</b>	<b>getprotobynumber</b>	<b>getprotoent</b>																																																																														
<b>getservbyname</b>	<b>getservbyport</b>	<b>getservent</b>																																																																														
<b>getsockname</b>	<b>getsockopt</b>	<b>h_errno</b>																																																																														
<b>htonl</b>	<b>htons</b>	<b>inet_addr</b>																																																																														
<b>inet_lnaof</b>	<b>inet_makeaddr</b>	<b>inet_netof</b>																																																																														
<b>inet_network</b>	<b>inet_ntoa</b>	<b>listen</b>																																																																														
<b>ntohl</b>	<b>ntohs</b>	<b>recv</b>																																																																														
<b>recvfrom</b>	<b>recvmsg</b>	<b>send</b>																																																																														
<b>sendmsg</b>	<b>sendto</b>	<b>sethostent</b>																																																																														
<b>setnetent</b>	<b>setprotoent</b>	<b>setservent</b>																																																																														
<b>setsockopt</b>	<b>shutdown</b>	<b>socket</b>																																																																														
<b>socketpair</b>	<b>t_accept</b>	<b>t_alloc</b>																																																																														
<b>t_bind</b>	<b>t_close</b>	<b>t_connect</b>																																																																														
<b>t_errno</b>	<b>t_error</b>	<b>t_free</b>																																																																														
<b>t_getinfo</b>	<b>t_getprotaddr</b>	<b>t_getstate</b>																																																																														
<b>t_listen</b>	<b>t_look</b>	<b>t_open</b>																																																																														
<b>t_optmgmt</b>	<b>t_rcv</b>	<b>t_rcvconnect</b>																																																																														
<b>t_rcvdis</b>	<b>t_rcvrel</b>	<b>t_rcvudata</b>																																																																														
<b>t_rcvuderr</b>	<b>t_snd</b>	<b>t_snddis</b>																																																																														
<b>t_sndrel</b>	<b>t_sndudata</b>	<b>t_strerror</b>																																																																														
<b>t_sync</b>	<b>t_unbind</b>																																																																															
<b>FILES</b>	<b>/usr/lib/libxnet.so.1</b>	shared object																																																																														

**ATTRIBUTES**

See **attributes(5)** for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT-Level	Safe

**SEE ALSO**

**intro(3)**, **attributes(5)**, **standards(5)**

<b>NAME</b>	limits – header for implementation-specific constants	
<b>SYNOPSIS</b>	#include <limits.h>	
<b>DESCRIPTION</b>	The header <limits.h> is a list of minimal magnitude limitations imposed by a specific implementation of the operating system.	
	ARG_MAX	1048320 /* max length of arguments to exec */
	CHAR_BIT	8 /* max # of bits in a "char" */
	CHAR_MAX	255 /* max value of a "char" */
	CHAR_MIN	0 /* min value of a "char" */
	CHILD_MAX	25 /* max # of processes per user id */
	CLK_TCK	_sysconf(3) /* clock ticks per second */
	DBL_DIG	15 /* digits of precision of a "double" */
	DBL_MAX	1.7976931348623157E+308 /* max decimal value of a "double" */
	DBL_MIN	2.2250738585072014E-308 /* min decimal value of a "double" */
	FCHR_MAX	1048576 /* historical default file size limit in bytes */
	FLT_DIG	6 /* digits of precision of a "float" */
	FLT_MAX	3.40282347e+38F /* max decimal value of a "float" */
	FLT_MIN	1.17549435E-38F /* min decimal value of a "float" */
	INT_MAX	2147483647 /* max value of an "int" */
	INT_MIN	(-2147483647-1) /* min value of an "int" */
	LINK_MAX	1000 /* max # of links to a single file */
	LOGNAME_MAX	8 /* max # of characters in a login name */
	LONG_BIT	32 /* # of bits in a "long" */
	LONG_MAX	2147483647 /* max value of a "long int" */
	LONG_MIN	(-2147483647-1) /* min value of a "long int" */
	MAX_CANON	256 /* max bytes in a line for canonical processing */
	MAX_INPUT	512 /* max size of a char input buffer */
	MB_LEN_MAX	5 /* max # of bytes in a multibyte character */
	NAME_MAX	14 /* max # of characters in a file name */
	NGROUPS_MAX	16 /* max # of groups for a user */
	NL_ARGMAX	9 /* max value of "digit" in calls to the NLS printf() and scanf() */
	NL_LANGMAX	14 /* max # of bytes in a LANG name */
	NL_MSGMAX	32767 /* max message number */
	NL_NMAX	1 /* max # of bytes in N-to-1 mapping characters */
	NL_SETMAX	255 /* max set number */
	NL_TEXTMAX	255 /* max # of bytes in a message string */
	NZERO	20 /* default process priority */
	OPEN_MAX	20 /* max # of files a process can have open */
	PASS_MAX	8 /* max # of characters in a password */
	PATH_MAX	1024 /* max # of characters in a path name */
	PID_MAX	30000 /* max value for a process ID */

<b>PIPE_BUF</b>	<b>5120</b>	<i>/* max # bytes atomic in write to a pipe */</i>
<b>PIPE_MAX</b>	<b>5120</b>	<i>/* max # bytes written to a pipe in a write */</i>
<b>SCHAR_MAX</b>	<b>127</b>	<i>/* max value of a "signed char" */</i>
<b>SCHAR_MIN</b>	<b>(-128)</b>	<i>/* min value of a "signed char" */</i>
<b>SHRT_MAX</b>	<b>32767</b>	<i>/* max value of a "short int" */</i>
<b>SHRT_MIN</b>	<b>(-32768)</b>	<i>/* min value of a "short int" */</i>
<b>STD_BLK</b>	<b>1024</b>	<i>/* # bytes in a physical I/O block */</i>
<b>SYS_NMLN</b>	<b>257</b>	<i>/* 4.0 size of utsname elements */</i>
		<i>/* also defined in sys/utsname.h */</i>
<b>SYSPID_MAX</b>	<b>1</b>	<i>/* max pid of system processes */</i>
<b>TMP_MAX</b>	<b>17576</b>	<i>/* max # of unique names generated by tmpnam */</i>
<b>UCHAR_MAX</b>	<b>255</b>	<i>/* max value of an "unsigned char" */</i>
<b>UID_MAX</b>	<b>2147483647</b>	<i>/* max value for a user or group ID */</i>
<b>UINT_MAX</b>	<b>4294967295</b>	<i>/* max value of an "unsigned int" */</i>
<b>ULONG_MAX</b>	<b>4294967295</b>	<i>/* max value of an "unsigned long int" */</i>
<b>USHRT_MAX</b>	<b>65535</b>	<i>/* max value of an "unsigned short int" */</i>
<b>USI_MAX</b>	<b>4294967295</b>	<i>/* max decimal value of an "unsigned" */</i>
<b>WORD_BIT</b>	<b>32</b>	<i>/* # of bits in a "word" or "int" */</i>

The following POSIX definitions are the most restrictive values to be used by a POSIX-conforming application (see **standards(5)**). Conforming implementations shall provide values at least this large.

<b>_POSIX_ARG_MAX</b>	<b>4096</b>	<i>/* max length of arguments to exec */</i>
<b>_POSIX_CHILD_MAX</b>	<b>6</b>	<i>/* max # of processes per user ID */</i>
<b>_POSIX_LINK_MAX</b>	<b>8</b>	<i>/* max # of links to a single file */</i>
<b>_POSIX_MAX_CANON</b>	<b>255</b>	<i>/* max # of bytes in a line of input */</i>
<b>_POSIX_MAX_INPUT</b>	<b>255</b>	<i>/* max # of bytes in terminal input queue */</i>
<b>_POSIX_NAME_MAX</b>	<b>14</b>	<i>/* # of bytes in a filename */</i>
<b>_POSIX_NGROUPS_MAX</b>	<b>0</b>	<i>/* max # of groups in a process */</i>
<b>_POSIX_OPEN_MAX</b>	<b>16</b>	<i>/* max # of files a process can have open */</i>
<b>_POSIX_PATH_MAX</b>	<b>255</b>	<i>/* max # of characters in a pathname */</i>
<b>_POSIX_PIPE_BUF</b>	<b>512</b>	<i>/* max # of bytes atomic in write to a pipe */</i>

**SEE ALSO**

**standards(5)**

<b>NAME</b>	loadfont – format of a font file used as input to the loadfont utility
<b>DESCRIPTION</b>	<p>This section describes the format of files that can be used to change the font used by the console when using the <b>loadfont(1)</b> utility with the <b>-f</b> option.</p> <p>The format is compatible with the Binary Distribution Format version 2.1 as developed by Adobe Systems, Inc.; however, certain restrictions apply. Video cards, when used with the Solaris for x86 system in text mode, only accept constant width and constant height fonts in certain sizes.</p> <p>The <b>loadfont</b> utility also requires that there is a description of all 256 characters of the codeset used specified in the fontfile. Certain attributes are not used by <b>loadfont</b> but are maintained for compatibility purposes.</p>
<b>File Format</b>	<p>A <b>loadfont</b> input file is a plain ASCII file containing only printable characters (octal 40 through 176) and a carriage return at the end of each line.</p> <p>The information about a particular font should be contained in a single file. The file begins with information on the font in general, followed by the information and bitmaps for the individual characters. The file should contain bitmaps for all 256 characters, and each character should be of the same size.</p> <p>A font bitmap description file has the following general form, where each item is contained on a separate line of text in the file. Items on a line are separated by spaces:</p> <p style="padding-left: 40px;">One or more lines beginning with the word <b>COMMENT</b>. These lines can be used to add comments to the file and will be ignored by the <b>loadfont</b> program.</p> <p style="padding-left: 40px;">The word <b>STARTFONT</b> followed by the version number 2.1.</p> <p style="padding-left: 40px;">The word <b>FONT</b> followed by the full name of the font. The name may continue all the way to the end of the line, and may contain spaces.</p> <p style="padding-left: 40px;">The word <b>SIZE</b> followed by the point size of the characters, the x resolution, and the y resolution of the font. This line is not used by <b>loadfont</b> but it needs to be there for compatibility purposes.</p> <p style="padding-left: 40px;">The word <b>FONTBOUNDINGBOX</b> followed by the width in x, height in y, and the x and y displacement of the lower left-hand corner from the origin. Again, this line is not used by <b>loadfont</b> but it must be there for compatibility purposes.</p> <p style="padding-left: 40px;">Optionally, the word <b>STARTPROPERTIES</b> followed by the number of properties that follow. If present, the number needs to match the number of lines following this one before the occurrence of a line beginning with <b>ENDPROPERTIES</b>. These lines consist of a word for the property name followed by either an integer or string surrounded by double quotes. Properties named <b>FONT_ASCENT</b>, <b>FONT_DESCENT</b> and <b>DEFAULT_CHAR</b> are typically present in BDF files to define the logical font-ascent and font-descent and the default-char for the font.</p> <p style="padding-left: 40px;">As mentioned above, this section, if it exists, must be terminated by <b>ENDPROPERTIES</b>.</p>

The word **CHARS** followed by the number of characters that follow. This number should always be **256**.

This terminates the part of the **loadfont** input file describing features of the font in general. The rest of the file contains descriptions of the individual characters. They consist of the following parts:

The word **STARTCHAR** followed by up to 14 characters (no blanks) describing the character. This can either be something like **C0041**, which indicates the hex value of the character or **uppercaseA**, which describes the character.

The word **ENCODING** followed by a positive integer representing value by which this character is represented internally in the codeset for which this font is used. The integer needs to be specified in decimal.

The word **SWIDTH** followed by the scalable width in x and y of character. Scalable widths are in units of 1/1000th of the size of the character. The y value should always be **0**; the x value is typically **666** for the type of characters used with **loadfont**. The values are not checked by the **loadfont** utility, but this line needs to be there for compatibility purposes.

The word **DWIDTH** followed by two numbers, which in a **BDF** file would mean the width in x and y of the character in device units. The y value is always zero. The x value is typically **8**. **loadfont** checks only for the presence of the **DWIDTH** keyword.

The word **BBX** followed by the width in x, height in y and x and y displacement of the lower left-hand corner from the origin of the character.

Most fonts used by video cards will not use the bottom 4 rows of pixels, which basically means a vertical (y) displacement of **-4**. The only width allowed by **loadfont** is **8**; heights supported are **8**, **14**, and **16**. All **BBX** lines of the subsequent characters should list the same height and width as the first one (because only fixed size fonts are supported).

The optional word **ATTRIBUTES** followed by the attributes as 4 hex-encoded characters. The **loadfont** utility will accept this line, if present, but there is no meaning attached to it.

The word **BITMAP**, which indicates the beginning of the bitmap representation of the character. This line should be followed by **height** number of lines (height as specified in the **BBX** line) representing a hex-encoded bitmap of the character, one byte per line.

The word **ENDCHAR** indicating the end of the bitmap for this character.

After all the bitmaps, the end of the file is indicated by the **ENDFONT** keyword.

**Example**

The following example lists the beginning of the **loadfont** input file for an 8 by 16 font, supporting the IBM 437 codeset, as well as the bitmap representation of the character uppercase A.

```

STARTFONT 2.1
FONT 8x16
SIZE 16 75 75
FONTBOUNDINGBOX 8 16 0 -4
STARTPROPERTIES 3
FONT_DESCENT 4
FONT_ASCENT 12
DEFAULT_CHAR 0
ENDPROPERTIES
CHARS 256
STARTCHAR C0000
ENCODING 0
...

```

Bitmap for uppercase A character:

```

STARTCHAR C0041
ENCODING 65
SWIDTH 666 0
DWIDTH 8 0
BBX 8 16 0 -4
BITMAP
00
00
10
38
6c
c6
c6
fe
c6
c6
c6
c6
c6
00
00
00
00
ENDCHAR

```

**FILES** /usr/share/lib/\*.bdf

**ATTRIBUTES** See **attributes(5)** for descriptions of the following attributes:



ATTRIBUTE TYPE	ATTRIBUTE VALUE
Architecture	x86

**SEE ALSO** [loadfont\(1\)](#), [attributes\(5\)](#)

<b>NAME</b>	logindevperm, fctab – login-based device permissions
<b>SYNOPSIS</b>	<b>/etc/logindevperm</b>
<b>DESCRIPTION</b>	<p>The <b>/etc/logindevperm</b> file contains information that is used by <b>login(1)</b> and <b>ttymon(1M)</b> to change the owner, group, and permissions of devices upon logging into or out of a console device. By default, this file contains lines for the keyboard, mouse, audio, and frame buffer devices.</p> <p>The owner of the devices listed in <b>/etc/logindevperm</b> is set to the owner of the console by <b>login(1)</b>. The group of the devices is set to the owner's group specified in <b>/etc/passwd</b>. The permissions are set as specified in <b>/etc/logindevperm</b>.</p> <p>Fields are separated by TAB and/or SPACE characters. Blank lines and comments can appear anywhere in the file; comments start with a hashmark, '#', and continue to the end of the line.</p> <p>The first field specifies the name of a console device (for example, <b>/dev/console</b>). The second field specifies the permissions to which the devices in the <i>device_list</i> field (third field) will be set. A <i>device_list</i> is a colon-separated list of device names. A device entry that is a directory name and ends with "/*" specifies all entries in the directory (except "." and ".."). For example, <b>"/dev/fbs/*"</b> specifies all frame buffer devices.</p> <p>Once the devices are owned by the user, their permissions and ownership can be changed using <b>chmod(1)</b> and <b>chown(1)</b>, as with any other user-owned file.</p> <p>Upon logout the owner and group of these devices will be reset by <b>ttymon(1M)</b> to owner <b>root</b> and root's group as specified in <b>/etc/passwd</b> (typically <b>other</b>). The permissions are set as specified in the <b>/etc/logindevperm</b> file.</p>
<b>FILES</b>	<b>/etc/passwd</b> File that contains user group information.
<b>SEE ALSO</b>	<b>chmod(1)</b> , <b>chown(1)</b> , <b>login(1)</b> , <b>ttymon(1M)</b> , <b>passwd(4)</b>
<b>NOTES</b>	<b>/etc/logindevperm</b> provides a superset of the functionality provided by <b>/etc/fctab</b> in SunOS 4.x releases.

<b>NAME</b>	loginlog – log of failed login attempts
<b>DESCRIPTION</b>	<p>After five unsuccessful login attempts, all the attempts are logged in the file <b>/var/adm/loginlog</b>. This file contains one record for each failed attempt. Each record contains the login name, tty specification, and time.</p> <p>This is an ASCII file. Each field within each entry is separated from the next by a colon. Each entry is separated from the next by a new-line.</p> <p>By default, <b>loginlog</b> does not exist, so no logging is done. To enable logging, the log file must be created with read and write permission for owner only. Owner must be <b>root</b> and group must be <b>sys</b>.</p>
<b>FILES</b>	<b>/var/adm/loginlog</b>
<b>SEE ALSO</b>	<b>login(1), passwd(1)</b>

<b>NAME</b>	magic – file command's magic number file																																																
<b>SYNOPSIS</b>	<code>/etc/magic</code>																																																
<b>DESCRIPTION</b>	<p>The <b>file</b>(1) command identifies the type of a file using, among other tests, a test for whether the file begins with a certain <i>magic number</i>. The <code>/etc/magic</code> file specifies what magic numbers are to be tested for, what message to print if a particular magic number is found, and additional information to extract from the file.</p> <p>Each line of the file specifies a test to perform. A test compares the data starting at a particular offset in the file with a 1-byte, 2-byte, or 4-byte numeric value or a string. If the test succeeds, a message is printed. The line consists of the following fields (separated by tabs):</p> <table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;"><i>offset</i></th> <th style="text-align: left;"><i>type</i></th> <th style="text-align: left;"><i>value</i></th> <th style="text-align: left;"><i>message</i></th> </tr> </thead> <tbody> <tr> <td><i>offset</i></td> <td colspan="3">A number specifying the offset, in bytes, into the file of the data which is to be tested.</td> </tr> <tr> <td><i>type</i></td> <td colspan="3">The type of the data to be tested. The possible values are:</td> </tr> <tr> <td></td> <td><b>byte</b></td> <td colspan="2">A one-byte value.</td> </tr> <tr> <td></td> <td><b>short</b></td> <td colspan="2">A two-byte value.</td> </tr> <tr> <td></td> <td><b>long</b></td> <td colspan="2">A four-byte value.</td> </tr> <tr> <td></td> <td><b>string</b></td> <td colspan="2">A string of bytes.</td> </tr> <tr> <td></td> <td colspan="3">The types <b>byte</b>, <b>short</b>, and <b>long</b> may optionally be followed by a mask specifier of the form <i>&amp;number</i>. If a mask specifier is given, the value is AND'ed with the <i>number</i> before any comparisons are done. The <i>number</i> is specified in C form. For instance, <b>13</b> is decimal, <b>013</b> is octal, and <b>0x13</b> is hexadecimal.</td> </tr> <tr> <td><i>value</i></td> <td colspan="3">The value to be compared with the value from the file. If the type is numeric, this value is specified in C form. If it is a string, it is specified as a C string with the usual escapes permitted (for instance, <code>\n</code> for NEWLINE).</td> </tr> <tr> <td></td> <td colspan="3"><i>Numeric values</i> may be preceded by a character indicating the operation to be performed. It may be <code>'='</code>, to specify that the value from the file must equal the specified value, <code>'&lt;'</code>, to specify that the value from the file must be less than the specified value, <code>'&gt;'</code>, to specify that the value from the file must be greater than the specified value, <code>'&amp;'</code>, to specify that all the bits in the specified value must be set in the value from the file, <code>'^'</code>, to specify that at least one of the bits in the specified value must not be set in the value from the file, or <code>x</code> to specify that any value will match. If the character is omitted, it is assumed to be <code>'='</code>.</td> </tr> <tr> <td></td> <td colspan="3">For string values, the byte string from the file must match the specified byte string. The byte string from the file which is matched is the same length as the specified byte string.</td> </tr> <tr> <td><i>message</i></td> <td colspan="3">The message to be printed if the comparison succeeds. If the string contains a <b>printf</b>(3S) format specification, the value from the file (with any specified</td> </tr> </tbody> </table>	<i>offset</i>	<i>type</i>	<i>value</i>	<i>message</i>	<i>offset</i>	A number specifying the offset, in bytes, into the file of the data which is to be tested.			<i>type</i>	The type of the data to be tested. The possible values are:				<b>byte</b>	A one-byte value.			<b>short</b>	A two-byte value.			<b>long</b>	A four-byte value.			<b>string</b>	A string of bytes.			The types <b>byte</b> , <b>short</b> , and <b>long</b> may optionally be followed by a mask specifier of the form <i>&amp;number</i> . If a mask specifier is given, the value is AND'ed with the <i>number</i> before any comparisons are done. The <i>number</i> is specified in C form. For instance, <b>13</b> is decimal, <b>013</b> is octal, and <b>0x13</b> is hexadecimal.			<i>value</i>	The value to be compared with the value from the file. If the type is numeric, this value is specified in C form. If it is a string, it is specified as a C string with the usual escapes permitted (for instance, <code>\n</code> for NEWLINE).				<i>Numeric values</i> may be preceded by a character indicating the operation to be performed. It may be <code>'='</code> , to specify that the value from the file must equal the specified value, <code>'&lt;'</code> , to specify that the value from the file must be less than the specified value, <code>'&gt;'</code> , to specify that the value from the file must be greater than the specified value, <code>'&amp;'</code> , to specify that all the bits in the specified value must be set in the value from the file, <code>'^'</code> , to specify that at least one of the bits in the specified value must not be set in the value from the file, or <code>x</code> to specify that any value will match. If the character is omitted, it is assumed to be <code>'='</code> .				For string values, the byte string from the file must match the specified byte string. The byte string from the file which is matched is the same length as the specified byte string.			<i>message</i>	The message to be printed if the comparison succeeds. If the string contains a <b>printf</b> (3S) format specification, the value from the file (with any specified		
<i>offset</i>	<i>type</i>	<i>value</i>	<i>message</i>																																														
<i>offset</i>	A number specifying the offset, in bytes, into the file of the data which is to be tested.																																																
<i>type</i>	The type of the data to be tested. The possible values are:																																																
	<b>byte</b>	A one-byte value.																																															
	<b>short</b>	A two-byte value.																																															
	<b>long</b>	A four-byte value.																																															
	<b>string</b>	A string of bytes.																																															
	The types <b>byte</b> , <b>short</b> , and <b>long</b> may optionally be followed by a mask specifier of the form <i>&amp;number</i> . If a mask specifier is given, the value is AND'ed with the <i>number</i> before any comparisons are done. The <i>number</i> is specified in C form. For instance, <b>13</b> is decimal, <b>013</b> is octal, and <b>0x13</b> is hexadecimal.																																																
<i>value</i>	The value to be compared with the value from the file. If the type is numeric, this value is specified in C form. If it is a string, it is specified as a C string with the usual escapes permitted (for instance, <code>\n</code> for NEWLINE).																																																
	<i>Numeric values</i> may be preceded by a character indicating the operation to be performed. It may be <code>'='</code> , to specify that the value from the file must equal the specified value, <code>'&lt;'</code> , to specify that the value from the file must be less than the specified value, <code>'&gt;'</code> , to specify that the value from the file must be greater than the specified value, <code>'&amp;'</code> , to specify that all the bits in the specified value must be set in the value from the file, <code>'^'</code> , to specify that at least one of the bits in the specified value must not be set in the value from the file, or <code>x</code> to specify that any value will match. If the character is omitted, it is assumed to be <code>'='</code> .																																																
	For string values, the byte string from the file must match the specified byte string. The byte string from the file which is matched is the same length as the specified byte string.																																																
<i>message</i>	The message to be printed if the comparison succeeds. If the string contains a <b>printf</b> (3S) format specification, the value from the file (with any specified																																																

masking performed) is printed using the message as the format string.

Some file formats contain additional information which is to be printed along with the file type. A line which begins with the character '>' indicates additional tests and messages to be printed. If the test on the line preceding the first line with a '>' succeeds, the tests specified in all the subsequent lines beginning with '>' are performed, and the messages printed if the tests succeed. The next line which does not begin with a '>' terminates this.

**FILES** /etc/magic

**SEE ALSO** file(1), file(1B), printf(3S)

**BUGS** There should be more than one level of subtests, with the level indicated by the number of '>' at the beginning of the line.

<b>NAME</b>	<b>mnttab</b> – mounted file system table										
<b>DESCRIPTION</b>	<p>The file <b>mnttab</b> resides in <b>/etc</b> and contains information about devices that are <i>currently</i> mounted. <b>mnttab</b> is read by programs using the routines described in <b>getmntent(3C)</b>. <b>mount(1M)</b> adds entries to this file. <b>umount</b> removes entries from this file. Each entry is a line of fields separated by spaces in the form:</p> <p style="padding-left: 40px;"><i>special mount_point fstype options time</i></p> <p>where</p> <table border="0" style="padding-left: 40px;"> <tr> <td><i>special</i></td> <td>The name of the resource to be mounted.</td> </tr> <tr> <td><i>mount_point</i></td> <td>The pathname of the directory on which the filesystem is mounted.</td> </tr> <tr> <td><i>fstype</i></td> <td>The file system type of the mounted file system.</td> </tr> <tr> <td><i>options</i></td> <td>The mount options. (See repective mount file system man page below in <b>SEE ALSO</b>.)</td> </tr> <tr> <td><i>time</i></td> <td>The time at which the file system was mounted.</td> </tr> </table> <p>Examples of entries for the <i>special</i> field include the pathname of a block-special device, the name of a remote filesystem in <i>host:pathname</i> form, or the name of a “swap file” (for instance, a file made with <b>mkfile(1M)</b>).</p>	<i>special</i>	The name of the resource to be mounted.	<i>mount_point</i>	The pathname of the directory on which the filesystem is mounted.	<i>fstype</i>	The file system type of the mounted file system.	<i>options</i>	The mount options. (See repective mount file system man page below in <b>SEE ALSO</b> .)	<i>time</i>	The time at which the file system was mounted.
<i>special</i>	The name of the resource to be mounted.										
<i>mount_point</i>	The pathname of the directory on which the filesystem is mounted.										
<i>fstype</i>	The file system type of the mounted file system.										
<i>options</i>	The mount options. (See repective mount file system man page below in <b>SEE ALSO</b> .)										
<i>time</i>	The time at which the file system was mounted.										
<b>FILES</b>	<b>/etc/mnttab</b>										
<b>SEE ALSO</b>	<b>mkfile(1M)</b> , <b>mount_cachefs(1M)</b> , <b>mount_hfs(1M)</b> , <b>mount_nfs(1M)</b> , <b>mount_pcfs(1M)</b> , <b>mount_ufs(1M)</b> , <b>mount(1M)</b> , <b>setmnt(1M)</b> , <b>getmntent(3C)</b>										

<b>NAME</b>	netconfig – network configuration database								
<b>SYNOPSIS</b>	<code>/etc/netconfig</code>								
<b>DESCRIPTION</b>	<p>The network configuration database, <code>/etc/netconfig</code>, is a system file used to store information about networks that are connected to the system. The <code>netconfig</code> database and the routines that access it (see <code>getnetconfig(3N)</code>) are part of the Network Selection component. The Network Selection component also includes <code>getnetpath(3N)</code> routines to provide application-specific network search paths. These routines access the <code>netconfig</code> database based on the environment variable <code>NETPATH</code> (see <code>environ(5)</code>).</p> <p><code>netconfig</code> contains an entry for each network available on the system. Entries are separated by newlines. Fields are separated by whitespace and occur in the order in which they are described below. Whitespace can be embedded as “<code>\blank</code>” or “<code>\tab</code>”. Backslashes may be embedded as “<code>\\</code>”. Lines in <code>/etc/netconfig</code> that begin with a # (hash) in column 1 are treated as comments.</p> <p>Each of the valid lines in the <code>netconfig</code> database correspond to an available transport. Each entry is of the form:</p> <p><i>network ID semantics flag protocol-family protocol-name network-device translation-libraries</i></p> <p><i>network ID</i> A string used to uniquely identify a network. <i>network ID</i> consists of non-null characters, and has a length of at least 1. No maximum length is specified. This namespace is locally significant and the local system administrator is the naming authority. All <i>network IDs</i> on a system must be unique.</p> <p><i>semantics</i> The <i>semantics</i> field is a string identifying the “semantics” of the network, that is, the set of services it supports, by identifying the service interface it provides. The <i>semantics</i> field is mandatory. The following semantics are recognized.</p> <table border="0" style="margin-left: 40px;"> <tr> <td><b>tpi_clts</b></td> <td>Transport Provider Interface, connectionless</td> </tr> <tr> <td><b>tpi_cots</b></td> <td>Transport Provider Interface, connection oriented</td> </tr> <tr> <td><b>tpi_cots_ord</b></td> <td>Transport Provider Interface, connection oriented, supports orderly release.</td> </tr> </table> <p><i>flag</i> The <i>flag</i> field records certain two-valued (“true” and “false”) attributes of networks. <i>flag</i> is a string composed of a combination of characters, each of which indicates the value of the corresponding attribute. If the character is present, the attribute is “true.” If the character is absent, the attribute is “false.” “-” indicates that none of the attributes are present. Only one character is currently recognized:</p> <table border="0" style="margin-left: 40px;"> <tr> <td><b>v</b></td> <td>Visible (“default”) network. Used when the environment variable <code>NETPATH</code> is unset.</td> </tr> </table>	<b>tpi_clts</b>	Transport Provider Interface, connectionless	<b>tpi_cots</b>	Transport Provider Interface, connection oriented	<b>tpi_cots_ord</b>	Transport Provider Interface, connection oriented, supports orderly release.	<b>v</b>	Visible (“default”) network. Used when the environment variable <code>NETPATH</code> is unset.
<b>tpi_clts</b>	Transport Provider Interface, connectionless								
<b>tpi_cots</b>	Transport Provider Interface, connection oriented								
<b>tpi_cots_ord</b>	Transport Provider Interface, connection oriented, supports orderly release.								
<b>v</b>	Visible (“default”) network. Used when the environment variable <code>NETPATH</code> is unset.								

*protocol family*

The *protocol family* and *protocol name* fields are provided for protocol-specific applications.

The *protocol family* field contains a string that identifies a protocol family. The *protocol family* identifier follows the same rules as those for *network IDs*; the string consists of non-null characters, it has a length of at least 1, and there is no maximum length specified. A “-” in the *protocol family* field indicates that no protocol family identifier applies (the network is experimental). The following are examples:

<b>loopback</b>	Loopback (local to host).
<b>inet</b>	Internetwork: UDP, TCP, etc.
<b>implink</b>	ARPANET imp addresses
<b>pup</b>	PUP protocols: for example, BSP
<b>chaos</b>	MIT CHAOS protocols
<b>ns</b>	XEROX NS protocols
<b>nbs</b>	NBS protocols
<b>ecma</b>	European Computer Manufacturers Association
<b>datakit</b>	DATAKIT protocols
<b>ccitt</b>	CCITT protocols, X.25, etc.
<b>sna</b>	IBM SNA
<b>decnet</b>	DECNET
<b>dli</b>	Direct data link interface
<b>lat</b>	LAT
<b>hylink</b>	NSC Hyperchannel
<b>appletalk</b>	Apple Talk
<b>nit</b>	Network Interface Tap
<b>ieee802</b>	IEEE 802.2; also ISO 8802
<b>osi</b>	Umbrella for all families used by OSI (for example, protosw lookup)
<b>x25</b>	CCITT X.25 in particular
<b>osinet</b>	AFI = 47, IDI = 4
<b>gosip</b>	U.S. Government OSI

*protocol name* The *protocol name* field contains a string that identifies a protocol. The *protocol name* identifier follows the same rules as those for *network IDs*; that is, the string consists of non-NULL characters, it has a length of at least 1, and there is no maximum length specified. A “-” indicates that none of the names listed apply. The following protocol names are recognized.

<b>tcp</b>	Transmission Control Protocol
<b>udp</b>	User Datagram Protocol
<b>icmp</b>	Internet Control Message Protocol



*network device*

The *network device* is the full pathname of the device used to connect to the transport provider. Typically, this device will be in the `/dev` directory. The *network device* must be specified.

*translation libraries*

The *name-to-address translation libraries* support a “directory service” (a name-to-address mapping service) for the network. A “-” in this field indicates the absence of any *translation libraries*. This has a special meaning for networks of the protocol family **inet**: its name-to-address mapping is provided by the name service switch based on the entries for **hosts** and **services** in **nsswitch.conf(4)**. For networks of other families, a “-” indicates non-functional name-to-address mapping. Otherwise, this field consists of a comma-separated list of pathnames to dynamically linked libraries. The pathname of the library can be either absolute or relative. See **dlopen(3X)**.

Each field corresponds to an element in the **struct netconfig** structure. **struct netconfig** and the identifiers described on this manual page are defined in `<netconfig.h>`. This structure includes the following members:

<b>char</b> * <i>nc_netid</i>	Network ID, including NULL terminator.
<b>unsigned long</b> <i>nc_semantics</i>	Semantics.
<b>unsigned long</b> <i>nc_flag</i>	Flags.
<b>char</b> * <i>nc_protofmly</i>	Protocol family.
<b>char</b> * <i>nc_proto</i>	Protocol name.
<b>char</b> * <i>nc_device</i>	Full pathname of the network device.
<b>unsigned long</b> <i>nc_nlookups</i>	Number of directory lookup libraries.
<b>char</b> ** <i>nc_lookups</i>	Names of the name-to-address translation libraries.
<b>unsigned long</b> <i>nc_unused[9]</i>	Reserved for future expansion.

The *nc\_semantics* field takes the following values, corresponding to the semantics identified above:

```
NC_TPI_CLTS
NC_TPI_COTS
NC_TPI_COTS_ORD
```

The *nc\_flag* field is a bitfield. The following bit, corresponding to the attribute identified above, is currently recognized. NC\_NOFLAG indicates the absence of any attributes.

```
NC_VISIBLE
```

**EXAMPLES**

Below is a sample **netconfig** file:

```
#
# The "Network Configuration" File.
#
# Each entry is of the form:
#
# <network_id> <semantics> <flags> <protofamily> <protoname> <device> \
#   <nametoaddr_libs>
#
# The "-" in <nametoaddr_libs> for inet family transports indicates
# redirection to the name service switch policies for "hosts" and
# "services". The "-" may be replaced by nametoaddr libraries that
# comply with the SVr4 specs, in which case the name service switch
# will not be used for netdir_getbyname, netdir_getbyaddr,
# gethostbyname, gethostbyaddr, getservbyname, and getservbyport.
# There are no nametoaddr_libs for the inet family in Solaris anymore.
#
udp      tpi_clts      v  inet      udp  /dev/udp  -
tcp      tpi_cots_ord  v  inet      tcp  /dev/tcp  -
rawip    tpi_raw       -  inet      -    /dev/rawip -
ticlts   tpi_clts      v  loopback  -    /dev/ticlts straddr.so
ticotsord tpi_cots_ord v  loopback  -    /dev/ticotsord straddr.so
ticots   tpi_cots      v  loopback  -    /dev/ticots  straddr.so
```

**FILES**

<netconfig.h>

**SEE ALSO**

**dlopen(3X)**, **getnetconfig(3N)**, **getnetpath(3N)**, **nsswitch.conf(4)**

*NFS Administration Guide*

*Transport Interfaces Programming Guide*

<b>NAME</b>	netgroup – list of network groups
<b>SYNOPSIS</b>	<b>/etc/netgroup</b>
<b>DESCRIPTION</b>	<p>A <b>netgroup</b> defines a network-wide group of hosts and users. Netgroups may be used to restrict access to shared NFS filesystems and for restricting remote login and shell access.</p> <p>Network groups are stored in one of the Network Information Services, either NIS or NIS+, not in a local file.</p> <p>This manual page describes the format for a file that may be used to supply input to the <b>makedbm</b>(1M) or <b>nisaddent</b>(1M) programs that are used to build the NIS map or NIS+ table, respectively.</p> <p>Each line of the file defines the name and membership of network group. The line should have the format:</p> <p style="padding-left: 40px;"><i>groupname member ...</i></p> <p>The items on a line may be separated by a combination of one or more spaces or tabs. The <i>groupname</i> is the name of the group being defined. This is followed by a list of members of the group. Each <i>member</i> is either another group name, all of whose members are to be included in the group being defined, or a triple of the form:</p> <p style="padding-left: 40px;"><i>(hostname,username,domainname)</i></p> <p>In each triple, any of the three fields <i>hostname</i>, <i>username</i>, and <i>domainname</i>, can be empty. An empty field signifies a "wildcard" matching any value in that field. Thus:</p> <p style="padding-left: 40px;"><b>everything (, ,this.domain)</b></p> <p>defines a group named "everything" for the domain "this.domain" to which every host and user belongs.</p> <p>The <i>domainname</i> field refers to the domain in which the triple is valid, not the domain containing the host or user.</p> <p>Netgroups can be used to control NFS mount access (see <b>share_nfs</b>(1M)) and to control remote login and shell access (see <b>hosts.equiv</b>(4)). They can also be used to control local login access (see <b>passwd</b>(4), <b>shadow</b>(4), and "compat" in <b>nsswitch.conf</b>(4)).</p> <p>When used for these purposes, a host is considered a member of a netgroup if the netgroup contains any triple in which the <i>hostname</i> field matches the name of the host <i>requesting</i> access and the <i>domainname</i> field matches the domain of the host <i>controlling</i> access.</p> <p>Similarly, a user is considered a member of a netgroup if the netgroup contains any triple in which the <i>username</i> field matches the name of the user requesting access and the</p>

*domainname* field matches the domain of the host controlling access.

Note that when netgroups are used to control NFS mount access, access is granted depending only on whether the requesting host is a member of the netgroup. Remote login and shell access can be controlled both on the basis of host and user membership in separate netgroups.

**FILES**     /etc/netgroup       used by /var/yp/Makefile on NIS masters to build the NIS netgroup map

Note that the netgroup information must always be stored in a network information service, either NIS or NIS+. The local file is only used to construct the netgroup NIS maps or NIS+ table; it is never consulted directly.

**SEE ALSO**   nis+(1), makedbm(1M), nisaddent(1M), share\_nfs(1M), innnetgr(3N), hosts(4), hosts.equiv(4), nsswitch.conf(4), passwd(4), shadow(4)

**NOTES**     **netgroup** requires NIS or NIS+.

Applications may make general membership tests using the **innnetgr()** function (see **innnetgr(3N)**).

Because the "-" character will not match any specific username or hostname, it is commonly used as a placeholder that will match only wildcarded membership queries. So, for example:

```

onlyhosts      (host1,-,our.domain) (host2,-,our.domain)
onlyusers      (-,john,our.domain) (-,linda,our.domain)

```

effectively define netgroups containing only hosts and only users, respectively. Any other string that is guaranteed not to be a legal username or hostname will also suffice for this purpose.

When a machine with multiple interfaces and multiple names is defined as a member of a netgroup, one must list all of the names (see **hosts(4)**). A manageable way to do this is to define a netgroup containing all of the machine names. For example, for a host "gateway" that has names "gateway-subnet1" and "gateway-subnet2" one may define the netgroup:

```
gateway (gateway-subnet1, ,our.domain) (gateway-subnet2, ,our.domain)
```

and use this netgroup **gateway** whenever the host is to be included in another netgroup.

<b>NAME</b>	netid – netname database								
<b>SYNOPSIS</b>	<b>/etc/netid</b>								
<b>DESCRIPTION</b>	<p>The <b>netid</b> file is a local source of information on mappings between netnames (see <b>secure_rpc(3N)</b>) and user ids or hostnames in the local domain. The <b>netid</b> file can be used in conjunction with, or instead of, the network source: NIS or NIS+. The <b>publickey</b> entry in the <b>nsswitch.conf</b> (see <b>nsswitch.conf(4)</b>) file determines which of these sources will be queried by the system to translate netnames to local user ids or hostnames.</p> <p>Each entry in the <b>netid</b> file is a single line of the form:</p> <pre>netname      uid:gid, gid, gid...</pre> <p>or</p> <pre>netname      0:hostname</pre> <p>The first entry associates a local user id with a netname. The second entry associates a hostname with a netname.</p> <p>The <b>netid</b> file field descriptions are as follows:</p> <table border="0"> <tr> <td style="vertical-align: top;"><i>netname</i></td> <td>The operating system independent network name for the user or host. <i>netname</i> has one of two formats. The format used to specify a host is of the form: <b>unix.hostname@domain</b> where <i>hostname</i> is the name of the host and <i>domain</i> is the network domain name. The format used to specify a user id is of the form: <b>unix.uid@domain</b> where <i>uid</i> is the numerical id of the user and <i>domain</i> is the network domain name.</td> </tr> <tr> <td style="vertical-align: top;"><i>uid</i></td> <td>The numerical id of the user (see <b>passwd(4)</b>). When specifying a host name, <i>uid</i> is always zero.</td> </tr> <tr> <td style="vertical-align: top;"><i>group</i></td> <td>The numerical id of the group the user belongs to (see <b>group(4)</b>). Several groups, separated by commas, may be listed for a single <i>uid</i>.</td> </tr> <tr> <td style="vertical-align: top;"><i>hostname</i></td> <td>The local hostname (see <b>hosts(4)</b>).</td> </tr> </table> <p>Blank lines are ignored. Any part of a line to the right of a '#' symbol is treated as a comment.</p>	<i>netname</i>	The operating system independent network name for the user or host. <i>netname</i> has one of two formats. The format used to specify a host is of the form: <b>unix.hostname@domain</b> where <i>hostname</i> is the name of the host and <i>domain</i> is the network domain name. The format used to specify a user id is of the form: <b>unix.uid@domain</b> where <i>uid</i> is the numerical id of the user and <i>domain</i> is the network domain name.	<i>uid</i>	The numerical id of the user (see <b>passwd(4)</b> ). When specifying a host name, <i>uid</i> is always zero.	<i>group</i>	The numerical id of the group the user belongs to (see <b>group(4)</b> ). Several groups, separated by commas, may be listed for a single <i>uid</i> .	<i>hostname</i>	The local hostname (see <b>hosts(4)</b> ).
<i>netname</i>	The operating system independent network name for the user or host. <i>netname</i> has one of two formats. The format used to specify a host is of the form: <b>unix.hostname@domain</b> where <i>hostname</i> is the name of the host and <i>domain</i> is the network domain name. The format used to specify a user id is of the form: <b>unix.uid@domain</b> where <i>uid</i> is the numerical id of the user and <i>domain</i> is the network domain name.								
<i>uid</i>	The numerical id of the user (see <b>passwd(4)</b> ). When specifying a host name, <i>uid</i> is always zero.								
<i>group</i>	The numerical id of the group the user belongs to (see <b>group(4)</b> ). Several groups, separated by commas, may be listed for a single <i>uid</i> .								
<i>hostname</i>	The local hostname (see <b>hosts(4)</b> ).								
<b>EXAMPLES</b>	<p>Here is a sample <b>netid</b> file:</p> <pre>unix.789@West.Sun.COM      789:30,65 unix.123@Bldg_xy.Sun.COM   123:20,1521 unix.candlestick@campus1.bayarea.EDU  0:candlestick</pre>								

<b>FILES</b>	<b>/etc/group</b>	groups file
	<b>/etc/hosts</b>	hosts database
	<b>/etc/netid</b>	netname database
	<b>/etc/passwd</b>	password file
	<b>/etc/publickey</b>	public key database

**SEE ALSO** **netname2user(3N)**, **secure\_rpc(3N)**, **group(4)**, **hosts(4)**, **nsswitch.conf(4)**, **passwd(4)**, **publickey(4)**

<b>NAME</b>	netmasks – network mask database
<b>SYNOPSIS</b>	<code>/etc/inet/netmasks</code> <code>/etc/netmasks</code>
<b>DESCRIPTION</b>	<p>The <b>netmasks</b> file contains network masks used to implement IP subnetting. It supports both standard subnetting as specified in <i>RFC-950</i> and variable length subnetting as specified in <i>RFC-1519</i>. When using standard subnetting there should be a single line for each network that is subnetted in this file with the network number, any number of SPACE or TAB characters, and the network mask to use on that network. Network numbers and masks may be specified in the conventional IP <code>`.'</code> (dot) notation (like IP host addresses, but with zeroes for the host part). For example,</p> <pre style="margin-left: 40px;"><b>128.32.0.0      255.255.255.0</b></pre> <p>can be used to specify that the Class B network 128.32.0.0 should have eight bits of subnet field and eight bits of host field, in addition to the standard sixteen bits in the network field.</p> <p>When using variable length subnetting, the format is identical. However, there should be a line for each subnet with the first field being the subnet and the second field being the netmask that applies to that subnet. The users of the database, such as <code>ifconfig(1M)</code>, perform a lookup to find the longest possible matching mask. It is possible to combine the <i>RFC-950</i> and <i>RFC-1519</i> form of subnet masks in the netmasks file. For example,</p> <pre style="margin-left: 40px;"><b>128.32.0.0      255.255.255.0</b> <b>128.32.27.0     255.255.255.240</b> <b>128.32.27.16    255.255.255.240</b> <b>128.32.27.32    255.255.255.240</b> <b>128.32.27.48    255.255.255.240</b> <b>128.32.27.64    255.255.255.240</b> <b>128.32.27.80    255.255.255.240</b> <b>128.32.27.96    255.255.255.240</b> <b>128.32.27.112   255.255.255.240</b> <b>128.32.27.128   255.255.255.240</b> <b>128.32.27.144   255.255.255.240</b> <b>128.32.27.160   255.255.255.240</b> <b>128.32.27.176   255.255.255.240</b> <b>128.32.27.192   255.255.255.240</b> <b>128.32.27.208   255.255.255.240</b> <b>128.32.27.224   255.255.255.240</b> <b>128.32.27.240   255.255.255.240</b> <b>128.32.64.0     255.255.255.192</b></pre> <p>can be used to specify different netmasks in different parts of the 128.32.0.0 Class B network number. Addresses 128.32.27.0 through 128.32.27.255 have a subnet mask with 28 bits in the combined network and subnet fields (often referred to as the subnet field) and</p>

4 bits in the host field. Furthermore, addresses 128.32.64.0 through 128.32.64.63 have a 26 bits in the subnet field. Finally, all other addresses in the range 128.32.0.0 through 128.32.255.255 have a 24 bit subnet field.

Invalid entries are ignored.

**SEE ALSO**

**ifconfig(1M)**, **inet(7P)**

Postel, Jon, and Mogul, Jeff, *Internet Standard Subnetting Procedure*, RFC 950, Network Information Center, SRI International, Menlo Park, Calif., August 1985.

V. Fuller, T. Li, J. Yu, K. Varadhan, *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*, RFC 1519, Network Information Center, SRI International, Menlo Park, Calif., September 1993.

T. Pummill, B. Manning, *Variable Length Subnet Table For IPv4*, RFC 1878, Network Information Center, SRI International, Menlo Park, Calif., December 1995.

**NOTES**

**/etc/inet/netmasks** is the official SVr4 name of the **netmasks** file. The symbolic link **/etc/netmasks** exists for BSD compatibility.



<b>NAME</b>	netrc – file for ftp remote login data
<b>DESCRIPTION</b>	<p>The <b>.netrc</b> file contains data for logging in to a remote host over the network for file transfers by <b>ftp</b>(1). This file resides in the user's home directory on the machine initiating the file transfer. Its permissions should be set to disallow read access by group and others (see <b>chmod</b>(1)).</p> <p>The following tokens are recognized; they may be separated by SPACE, TAB, or NEWLINE characters:</p> <p><b>machine name</b> Identify a remote machine name. The auto-login process searches the <b>.netrc</b> file for a <b>machine</b> token that matches the remote machine specified on the <b>ftp</b> command line or as an <b>open</b> command argument. Once a match is made, the subsequent <b>.netrc</b> tokens are processed, stopping when the EOF is reached or another <b>machine</b> token is encountered.</p> <p><b>login name</b> Identify a user on the remote machine. If this token is present, the auto-login process will initiate a login using the specified name.</p> <p><b>password string</b> Supply a password. If this token is present, the auto-login process will supply the specified string if the remote server requires a password as part of the login process. Note: if this token is present in the <b>.netrc</b> file, <b>ftp</b> will abort the auto-login process if the <b>.netrc</b> is readable by anyone besides the user.</p> <p><b>account string</b> Supply an additional account password. If this token is present, the auto-login process will supply the specified string if the remote server requires an additional account password, or the auto-login process will initiate an <b>ACCT</b> command if it does not.</p> <p><b>macdef name</b> Define a macro. This token functions the same as <b>ftp macdef</b>. A macro is defined with the specified name; its contents begin with the next <b>.netrc</b> line and continue until a null line (consecutive NEWLINE characters) is encountered. If a macro named <b>init</b> is defined, it is automatically executed as the last step in the auto-login process.</p>
<b>EXAMPLES</b>	<p>A <b>.netrc</b> file containing the following line:</p> <p style="padding-left: 40px;"><b>machine ray login demo password mypassword</b></p> <p>allows an autologin to the machine <b>ray</b> using the login name <b>demo</b> with password <b>mypassword</b>.</p>

<b>FILES</b>	<code>~/.netrc</code>
<b>SEE ALSO</b>	<code>chmod(1)</code> , <code>ftp(1)</code> , <code>in.ftpd(1M)</code>

<b>NAME</b>	networks – network name database  <b>/etc/inet/networks</b> <b>/etc/networks</b>
<b>DESCRIPTION</b>	<p>The <b>networks</b> file is a local source of information regarding the networks which comprise the Internet. The <b>networks</b> file can be used in conjunction with, or instead of, other networks sources, including the NIS maps <b>networks.byname</b> and <b>networks.byaddr</b> and the NIS+ table <b>networks</b>. Programs use the <b>getnetbyname(3N)</b> routines to access this information.</p> <p>The network file has a single line for each network, with the following information:</p> <p style="padding-left: 40px;"><i>official-network-name    network-number    aliases</i></p> <p>Items are separated by any number of SPACE and/or TAB characters. A '#' indicates the beginning of a comment; characters up to the end of the line are not interpreted by routines which search the file. This file is normally created from the official network database maintained at the Network Information Control Center (NIC), though local changes may be required to bring it up to date regarding unofficial aliases and/or unknown networks.</p> <p>Network numbers may be specified in the conventional dot ('.') notation using the <b>inet_network</b> routine from the Internet address manipulation library, <b>inet(7P)</b>. Network names may contain any printable character other than a field delimiter, NEWLINE, or comment character.</p>
<b>SEE ALSO</b>	<b>getnetbyname(3N)</b> , <b>inet(3N)</b> , <b>nsswitch.conf(4)</b> , <b>inet(7P)</b>
<b>NOTES</b>	<b>/etc/inet/networks</b> is the official SVR4 name of the <b>networks</b> file. The symbolic link <b>/etc/networks</b> exists for BSD compatibility.

<b>NAME</b>	nisfiles – NIS+ database files and directory structure
<b>SYNOPSIS</b>	<code>/var/nis</code>
<b>DESCRIPTION</b>	<p>The Network Information Service Plus (NIS+) uses a memory based, replicated database. This database uses a set of files in the <code>/var/nis</code> directory for checkpointing to table storage and for maintaining a transaction log. Additionally, the NIS+ server and client use files in this directory to store binding and state information.</p> <p>The NIS+ service implements an authentication and authorization system that is built upon Secure RPC. In this implementation, the service uses a table named <code>cred.org_dir.domain-name</code> to store the public and private keys of principals that are authorized to access the NIS+ namespace. It stores group access information in the sub-domain <code>groups_dir.domain-name</code> as <i>group</i> objects. These two tables appear as files in the <code>/var/nis/data</code> directory on the NIS+ server.</p> <p>Unlike the previous versions of the network information service, in NIS+, the information in the tables is initially loaded into the service from the ASCII files on the server and then updated using NIS+ utilities (see <code>nistbladm(1)</code>). Some sites may wish to periodically regenerate the ASCII files for archival purposes. To do this, a script should be added in the <code>crontab(1)</code> of the server that lists these tables and creates the ASCII file from the result.</p> <p>Note: Except for the <code>NIS_COLDSTART</code> and <code>NIS_SHARED_DIRCACHE</code> file, no other files should be manipulated by commands such as <code>cp(1)</code>, <code>mv(1)</code> or <code>rm(1)</code>. The transaction log file keeps logs of all changes made, and hence the files cannot be manipulated independently.</p> <p>The files described below are stored in the <code>/var/nis</code> directory:</p> <p><b>NIS_COLDSTART</b> Contains NIS+ directory objects that are to be preloaded into the NIS+ cache at startup time. This file is usually created at NIS+ installation time. See <code>nisinit(1M)</code> or <code>nisclient(1M)</code>.</p> <p><b>NIS_SHARED_DIRCACHE</b> Contains the current cache of NIS+ bindings being maintained by the cache manager. The contents can be viewed with <code>nisshowcache(1M)</code>.</p> <p><b>client_info</b> Contains configuration information (preferred servers, options, etc.) for <code>nis_cachemgr(1M)</code> and (potentially) other NIS+ clients on the system. It is manipulated by the <code>nisprefadm(1M)</code> command.</p> <p><b>.pref_servers</b> A cached copy of preferred server information. It is maintained by <code>nis_cachemgr</code>. Do not edit this file manually.</p> <p><b>trans.log</b> Contains a transaction log that is maintained by the NIS+ service. It can be viewed using the <code>nislog(1M)</code> command. This file contains holes. Its apparent size may be a lot higher than its actual size. There is only one transaction log per server.</p> <p><b>data.dict</b> A dictionary that is used by the NIS+ database to locate its files. It is created by the default NIS+ database package.</p>

<b>data.dict.log</b>	The log file for the database dictionary. When the server is checkpointed (see the <b>-C</b> option of <b>nisping(1M)</b> ), this file will be deleted.
<b>data</b>	Contains databases that the server uses.
<b>data/root.object</b>	On root servers, this file contains a directory object that describes the root of the name space.
<b>data/parent.object</b>	On root servers, this file contains a directory object that describes the parent namespace. This file is created by the <b>nisinit(1M)</b> command.
<b>data/table_name</b>	For each table in the directory there is a file with the same name that stores the information about that table. If there are subdirectories within this directory, the database for the table is stored in the file, <i>table_name.subdirectory</i> .
<b>data/table_name.log</b>	Contains the database log for the table <i>table_name</i> . The log file maintains the state of individual transactions to each database. When a database has been checkpointed (that is, all changes have been made to the <i>data/table_name</i> stable storage), this log file will be deleted.  Currently, NIS+ does not automatically do checkpointing. The system administrator may want to do <b>nisping-C</b> operations periodically (such as, once a day) to checkpoint the log file. This can be done either through a <b>cron(1M)</b> job, or manually.
<b>data/root_dir</b>	On root servers, this file stores the database associated with the root directory. It is similar to other table databases. The corresponding log file is called <b>root_dir.log</b> .
<b>data/cred.org_dir</b>	Table containing the credentials of principals in this NIS+ domain.
<b>data/groups_dir</b>	Table containing the group authorization objects needed by NIS+ to authorize group access.
<b>data/serving_list</b>	Contains a list of all NIS+ directories that are being served by the NIS+ server on this server. When this server is added or deleted from any NIS+ directory object, this file is updated by the server.

**SEE ALSO**

**cp(1)**, **crontab(1)**, **mv(1)**, **nis(1)**, **nis\_cachemgr(1M)**, **niscat(1)**, **nismatch(1)**, **nistbladm(1)**, **rm(1)**, **cron(1M)**, **nisclient(1M)**, **nisinit(1M)**, **nislog(1M)**, **nisping(1M)**, **nisprefadm(1M)**, **nisshowcache(1M)**, **nis\_db(3N)**, **nis\_objects(3N)**

<b>NAME</b>	nologin – message displayed to users attempting to log on in the process of a system shutdown
<b>SYNOPSIS</b>	<b>/etc/nologin</b>
<b>DESCRIPTION</b>	<p>The <b>/etc/nologin</b> file contains the message displayed to users attempting to log on to a machine in the process of being shutdown. After displaying the contents of the <b>nologin</b> file, the <b>login</b> procedure terminates, preventing the user from logging onto the machine. This procedure is preferable to terminating a user's session by <b>shutdown</b> shortly after the user has logged on.</p> <p>Logins by super-user are not affected by this procedure.</p> <p>The message contained in the <b>nologin</b> file is editable by super-user. A typical <b>nologin</b> file contains a message similar to:</p> <p style="text-align: center;"><b>NO LOGINS: System going down in 10 minutes.</b></p>
<b>SEE ALSO</b>	<b>login(1), rlogin(1), telnet(1), shutdown(1M)</b>

<b>NAME</b>	note – specify legal annotations
<b>SYNOPSIS</b>	<b>/usr/lib/note</b>
<b>DESCRIPTION</b>	<p>Each file in this directory contains the <b>NOTE</b> (also <b>_NOTE</b>) annotations legal for a single tool. The name of the file, by convention, should be the tool vendor's stock name, followed by a hyphen, followed by the tool name. For example, for Sun's <b>lock_lint</b> tool the filename should be <b>SUNW-lock_lint</b>.</p> <p>The file should contain the names of the annotations understood by the tool, one per line. For example, if a tool understands the following annotations:</p> <pre>NOTE(NOT_REACHED) NOTE(MUTEX_PROTECTS_DATA(list_lock, list_head))</pre> <p>then its file in <b>/usr/lib/note</b> should contain the entries:</p> <pre>NOT_REACHED MUTEX_PROTECTS_DATA</pre> <p>Blank lines, and lines beginning with a pound (#), are ignored.</p> <p>While <b>/usr/lib/note</b> is the default directory tools search for such files, they can be made to search other directories instead simply by setting environment variable <b>NOTEPATH</b> to contain the paths, separated by colons, of directories to be searched, e.g., <b>/usr/mytool/note:/usr/lib/note</b>.</p>
<b>USAGE</b>	These files are used by such tools whenever they encounter <b>NOTEs</b> they do not understand. If a file in <b>/usr/lib/note</b> contains the annotation, then it is valid. If no such file contains the annotation, then the tool should issue a warning complaining that it might be invalid.
<b>ENVIRONMENT</b>	<b>NOTEPATH</b> specify paths to be searched for annotation files. Paths are separated by colons (“.”).
<b>SEE ALSO</b>	<b>NOTE(3X)</b>

<b>NAME</b>	nscd.conf – name service cache daemon configuration
<b>SYNOPSIS</b>	<i>/etc/nscd.conf</i>
<b>DESCRIPTION</b>	<p>The <b>nscd.conf</b> file contains the configuration information for <b>nscd(1M)</b>. Each line specifies either an <i>attribute</i> and a <i>value</i>, or an <i>attribute</i>, <i>cachename</i>, and a <i>value</i>. Fields are separated either by SPACE or TAB characters. A '#' (number sign) indicates the beginning of a comment; characters up to the end of the line are not interpreted by <b>nscd</b>.</p> <p><i>cachename</i> is represented by <b>hosts</b>, <b>passwd</b>, or <b>groups</b>.</p> <p><i>attribute</i> supports the following:</p> <p><b>logfile</b> <i>debug-file-name</i> Specifies name of the file to which debug info should be written. Use <b>/dev/tty</b> for standard output.</p> <p><b>debug-level</b> <i>value</i> Sets the debug level desired. <i>value</i> may range from <b>0</b> (the default) to <b>10</b>. Use of this option causes <b>nscd(1M)</b> to run in the foreground and not become a daemon. Note that the output of the debugging command is not likely to remain the same from release-to-release; scripts should <i>not</i> rely on its format.</p> <p><b>enable-cache</b> <i>cachename value</i> Enables or disables the specified cache. <i>value</i> may be either <b>yes</b> or <b>no</b>.</p> <p><b>positive-time-to-live</b> <i>cachename value</i> Sets the time-to-live for positive entries (successful queries) in the specified cache. <i>value</i> is in integer seconds. Larger values increase cache hit rates and reduce mean response times, but increase problems with cache coherence. Note that sites that push (update) NIS maps nightly can set the value to be the equivalent of 12 hours or more with very good performance implications.</p> <p><b>negative-time-to-live</b> <i>cachename value</i> Sets the time-to-live for negative entries (unsuccessful queries) in the specified cache. <i>value</i> is in integer seconds. Can result in significant performance improvements if there are several files owned by uids (user IDs) not in system databases; should be kept small to reduce cache coherency problems.</p> <p><b>suggested-size</b> <i>cachename value</i> Sets the suggested number of hash buckets in the specified cache. This parameter should be changed only if the number of entries in the cache exceeds the suggested size by more than a factor of four or five. Since this is the internal hash table size, <i>value</i> should remain a prime number for optimum efficiency.</p> <p><b>keep-hot-count</b> <i>cachename value</i> This attribute allows the administrator to set the number of entries</p>



**nscd(1M)** is to keep current in the specified cache. *value* is an integer number which should approximate the number of entries frequently used during the day.

**check-files** *cachename value*

Enables or disables checking the file belonging to the specified *cachename* for changes. If enabled (which is the default), changes in the corresponding file cause the cache to be invalidated within 10 seconds. Can be disabled if files are never modified for a slight performance boost, particularly over NFS. *value* may be either **yes** or **no**.

**SEE ALSO**

**nscd(1M)**, **group(4)**, **hosts(4)**, **passwd(4)**

**WARNINGS**

The **nscd.conf** interface is included in this release on an uncommitted basis only, and is subject to change or removal in a future minor release.

<b>NAME</b>	nsswitch.conf – configuration file for the name service switch																																												
<b>SYNOPSIS</b>	<code>/etc/nsswitch.conf</code>																																												
<b>DESCRIPTION</b>	<p>The operating system uses a number of "databases" of information about hosts, users (<b>passwd/shadow</b>), groups and so forth. Data for these can come from a variety of sources: host-names and host-addresses, for example, may be found in <code>/etc/hosts</code>, NIS, NIS+, or DNS. Zero or more sources may be used for each database; the sources and their lookup order are specified in the <code>/etc/nsswitch.conf</code> file.</p> <p>The following databases use the switch file:</p> <table border="0"> <thead> <tr> <th style="text-align: left;"><i>Database</i></th> <th style="text-align: left;"><i>Used by</i></th> </tr> </thead> <tbody> <tr> <td><b>aliases</b></td> <td><b>sendmail</b>(1M)</td> </tr> <tr> <td><b>automount</b></td> <td><b>automount</b>(1M)</td> </tr> <tr> <td><b>bootparams</b></td> <td><b>rpc.bootparamd</b>(1M)</td> </tr> <tr> <td><b>ethers</b></td> <td><b>ethers</b>(3N)</td> </tr> <tr> <td><b>group</b></td> <td><b>getgrnam</b>(3C)</td> </tr> <tr> <td><b>hosts</b></td> <td><b>gethostbyname</b>(3N) (See "<b>Interaction with netconfig</b>" below.)</td> </tr> <tr> <td><b>netgroup</b></td> <td><b>innetgr</b>(3N)</td> </tr> <tr> <td><b>netmasks</b></td> <td><b>ifconfig</b>(1M)</td> </tr> <tr> <td><b>networks</b></td> <td><b>getnetbyname</b>(3N)</td> </tr> <tr> <td><b>passwd</b></td> <td><b>getpwnam</b>(3C), <b>getspnam</b>(3C)</td> </tr> <tr> <td><b>protocols</b></td> <td><b>getprotobyname</b>(3N)</td> </tr> <tr> <td><b>publickey</b></td> <td><b>getpublickey</b>(3N), <b>secure_rpc</b>(3N)</td> </tr> <tr> <td><b>rpc</b></td> <td><b>getrpcbyname</b>(3N)</td> </tr> <tr> <td><b>sendmailvars</b></td> <td><b>sendmail</b>(1M)</td> </tr> <tr> <td><b>services</b></td> <td><b>getservbyname</b>(3N) (See "<b>Interaction with netconfig</b>" below.)</td> </tr> </tbody> </table> <p>The following sources may be used:</p> <table border="0"> <thead> <tr> <th style="text-align: left;"><i>Source</i></th> <th style="text-align: left;"><i>Uses</i></th> </tr> </thead> <tbody> <tr> <td><b>files</b></td> <td><code>/etc/hosts</code>, <code>/etc/passwd</code>, <code>/etc/shadow</code> and so forth</td> </tr> <tr> <td><b>nis</b></td> <td>NIS (YP)</td> </tr> <tr> <td><b>nisplus</b></td> <td>NIS+</td> </tr> <tr> <td><b>dns</b></td> <td>Valid only for <b>hosts</b>; uses the Internet Domain Name Service.</td> </tr> <tr> <td><b>compat</b></td> <td>Valid only for <b>passwd</b> and <b>group</b>; implements "+" and "-". (See "<b>Interaction with +/- syntax</b>" below.) The <b>compat</b> source may not be supported in future releases.</td> </tr> </tbody> </table> <p>There is an entry in <code>/etc/nsswitch.conf</code> for each database. Typically these entries will be simple, such as "protocols: files" or "networks: files nisplus". However, when multiple sources are specified, it is sometimes necessary to define precisely the circumstances under which each source will be tried. A source can return one of the following codes:</p>	<i>Database</i>	<i>Used by</i>	<b>aliases</b>	<b>sendmail</b> (1M)	<b>automount</b>	<b>automount</b> (1M)	<b>bootparams</b>	<b>rpc.bootparamd</b> (1M)	<b>ethers</b>	<b>ethers</b> (3N)	<b>group</b>	<b>getgrnam</b> (3C)	<b>hosts</b>	<b>gethostbyname</b> (3N) (See " <b>Interaction with netconfig</b> " below.)	<b>netgroup</b>	<b>innetgr</b> (3N)	<b>netmasks</b>	<b>ifconfig</b> (1M)	<b>networks</b>	<b>getnetbyname</b> (3N)	<b>passwd</b>	<b>getpwnam</b> (3C), <b>getspnam</b> (3C)	<b>protocols</b>	<b>getprotobyname</b> (3N)	<b>publickey</b>	<b>getpublickey</b> (3N), <b>secure_rpc</b> (3N)	<b>rpc</b>	<b>getrpcbyname</b> (3N)	<b>sendmailvars</b>	<b>sendmail</b> (1M)	<b>services</b>	<b>getservbyname</b> (3N) (See " <b>Interaction with netconfig</b> " below.)	<i>Source</i>	<i>Uses</i>	<b>files</b>	<code>/etc/hosts</code> , <code>/etc/passwd</code> , <code>/etc/shadow</code> and so forth	<b>nis</b>	NIS (YP)	<b>nisplus</b>	NIS+	<b>dns</b>	Valid only for <b>hosts</b> ; uses the Internet Domain Name Service.	<b>compat</b>	Valid only for <b>passwd</b> and <b>group</b> ; implements "+" and "-". (See " <b>Interaction with +/- syntax</b> " below.) The <b>compat</b> source may not be supported in future releases.
<i>Database</i>	<i>Used by</i>																																												
<b>aliases</b>	<b>sendmail</b> (1M)																																												
<b>automount</b>	<b>automount</b> (1M)																																												
<b>bootparams</b>	<b>rpc.bootparamd</b> (1M)																																												
<b>ethers</b>	<b>ethers</b> (3N)																																												
<b>group</b>	<b>getgrnam</b> (3C)																																												
<b>hosts</b>	<b>gethostbyname</b> (3N) (See " <b>Interaction with netconfig</b> " below.)																																												
<b>netgroup</b>	<b>innetgr</b> (3N)																																												
<b>netmasks</b>	<b>ifconfig</b> (1M)																																												
<b>networks</b>	<b>getnetbyname</b> (3N)																																												
<b>passwd</b>	<b>getpwnam</b> (3C), <b>getspnam</b> (3C)																																												
<b>protocols</b>	<b>getprotobyname</b> (3N)																																												
<b>publickey</b>	<b>getpublickey</b> (3N), <b>secure_rpc</b> (3N)																																												
<b>rpc</b>	<b>getrpcbyname</b> (3N)																																												
<b>sendmailvars</b>	<b>sendmail</b> (1M)																																												
<b>services</b>	<b>getservbyname</b> (3N) (See " <b>Interaction with netconfig</b> " below.)																																												
<i>Source</i>	<i>Uses</i>																																												
<b>files</b>	<code>/etc/hosts</code> , <code>/etc/passwd</code> , <code>/etc/shadow</code> and so forth																																												
<b>nis</b>	NIS (YP)																																												
<b>nisplus</b>	NIS+																																												
<b>dns</b>	Valid only for <b>hosts</b> ; uses the Internet Domain Name Service.																																												
<b>compat</b>	Valid only for <b>passwd</b> and <b>group</b> ; implements "+" and "-". (See " <b>Interaction with +/- syntax</b> " below.) The <b>compat</b> source may not be supported in future releases.																																												

<i>Status</i>	<i>Meaning</i>
<b>SUCCESS</b>	Requested database entry was found
<b>UNAVAIL</b>	Source is not responding or corrupted
<b>NOTFOUND</b>	Source responded "no such entry"
<b>TRYAGAIN</b>	Source is busy, might respond to retries

For each status code, two actions are possible:

<i>Action</i>	<i>Meaning</i>
<b>continue</b>	Try the next source in the list
<b>return</b>	Return now

The complete syntax of an entry is

```
<entry> ::= <database> ":" [<source> [<criteria>]]*
<criteria> ::= "[" <criterion>+ "]"
<criterion> ::= <status> "=" <action>
<status> ::= "success" | "notfound" | "unavail" | "tryagain"
<action> ::= "return" | "continue"
```

Each entry occupies a single line in the file. Lines that are blank, or that start with white space, are ignored. Everything on a line following a # character is also ignored; the # character can begin anywhere in a line, to be used to begin comments. The <database> and <source> names are case-sensitive, but <action> and <status> names are case-insensitive.

The library functions contain compiled-in default entries that are used if the appropriate entry in **nsswitch.conf** is absent or syntactically incorrect.

The default criteria are to continue on anything except **SUCCESS**; in other words, [**SUCCESS**=return **NOTFOUND**=continue **UNAVAIL**=continue **TRYAGAIN**=continue].

The default, or explicitly specified, criteria are meaningless following the last source in an entry; and they are ignored, since the action is always to return to the caller irrespective of the status code the source returns.

#### Interaction with netconfig

In order to ensure that they all return consistent results, **gethostbyname(3N)**, **getservbyname(3N)**, and **netdir\_getbyname(3N)** functions are all implemented in terms of the same internal library function. This function obtains the system-wide source lookup policy for **hosts** and **services** based on the **inet** family entries in **netconfig(4)** and uses the switch entries only if the netconfig entries have a "-" in the last column for name-toaddr libraries. See the **NOTES** section in **gethostbyname(3N)** and **getservbyname(3N)** for details.

#### Interaction with FNS

When **gethostbyname(3N)**, **gethostbyname\_r(3N)**, or **netdir\_getbyname(3N)** are given a slash-separated FNS host name to look up (see **fns(5)** and **fns\_policies(5)**), then the host is looked up using FNS directly and **nsswitch.conf** is not consulted.

**Interaction with NIS+ NIS/YP-compatibility Mode**

The NIS+ server can be run in "YP-compatibility mode", where it handles NIS (YP) requests as well as NIS+ requests. In this case, the clients get much the same results (except for **getspnam(3C)**) from the "nis" source as from "nisplus"; however, "nisplus" is recommended instead of "nis".

**Interaction with server in DNS-forwarding Mode**

The NIS (YP) server can be run in "DNS-forwarding mode", where it forwards lookup requests to DNS for host-names and -addresses that do not exist in its database. In this case, specifying "nis" as a source for "hosts" is sufficient to get DNS lookups; "dns" need not be specified explicitly as a source.

Since SunOS 5.3 (Solaris 2.3), the NIS+ server in "NIS/YP-compatibility mode" can also be run in "DNS-forwarding mode" (see **rpc.nisd(1M)**). Forwarding is effective only for requests originating from its YP clients; "hosts" policy on these clients should be configured appropriately.

**Interaction with Password Aging**

When password aging is turned on, only a limited set of possible name services are permitted for the **passwd:** database in the **/etc/nsswitch.conf** file:

```
passwd:      files
passwd:      files nis
passwd:      files nisplus
passwd:      compat
passwd:      compat
passwd_compat: nisplus
```

Any other settings will cause the **passwd(1)** command to fail when it attempts to change the password after expiration and will prevent the user from logging in. These are the *only* permitted settings when password aging has been turned on. Otherwise, you can work around incorrect **passwd:** lines by using the **-r repository** argument to the **passwd(1)** command and using **passwd -r repository** to override the **nsswitch.conf** settings and specify in which name service you want to modify your password.

**Interaction with +/- syntax**

Releases prior to SunOS 5.0 did not have the name service switch but did allow the user some policy control. In **/etc/passwd** one could have entries of the form *+user* (include the specified user from NIS **passwd.byname**), *-user* (exclude the specified user) and *+* (include everything, except excluded users, from NIS **passwd.byname**). The desired behavior was often "everything in the file followed by everything in NIS", expressed by a solitary *+* at the end of **/etc/passwd**. The switch provides an alternative for this case ("passwd: files nis") that does not require *+* entries in **/etc/passwd** and **/etc/shadow** (the latter is a new addition to SunOS 5.0, see **shadow(4)**).

If this is not sufficient, the NIS/YP compatibility source provides full +/- semantics. It reads **/etc/passwd** for **getpwnam(3C)** functions and **/etc/shadow** for **getspnam(3C)** functions and, if it finds +/- entries, invokes an appropriate source. By default, the source is "nis", but this may be overridden by specifying "nisplus" as the source for the pseudo-database **passwd\_compat**.

### Useful Configurations

Note that for every `/etc/passwd` entry, there should be a corresponding entry in the `/etc/shadow` file.

The NIS/YP compatibility source also provides full +/- semantics for **group**; the relevant pseudo-database is **group\_compat**.

The compiled-in default entries for all databases use NIS (YP) as the enterprise level name service and are identical to those in the default configuration of this file:

```
passwd:      files nis
group:       files nis
hosts:       nis [NOTFOUND=return] files
networks:    nis [NOTFOUND=return] files
protocols:   nis [NOTFOUND=return] files
rpc:         nis [NOTFOUND=return] files
ethers:      nis [NOTFOUND=return] files
netmasks:   nis [NOTFOUND=return] files
bootparams:  nis [NOTFOUND=return] files
publickey:   nis [NOTFOUND=return] files
netgroup:    nis
automount:   files nis
aliases:     files nis
services:    files nis
sendmailvars: files
```

The policy "nis [NOTFOUND=return] files" implies "if **nis** is UNAVAIL, continue on to **files**, and if **nis** returns NOTFOUND, return to the caller; in other words, treat **nis** as the authoritative source of information and try **files** only if **nis** is down." This, and other policies listed in the default configuration above, are identical to the hard-wired policies in SunOS releases prior to 5.0.

If compatibility with the +/- syntax for **passwd** and **group** is required, simply modify the entries for **passwd** and **group** to:

```
passwd:      compat
group:       compat
```

If NIS+ is the enterprise level name service, the default configuration should be modified to use **nisplus** instead of **nis** for every database on client machines. The file `/etc/nsswitch.nisplus` contains a sample configuration that can be copied to `/etc/nsswitch.conf` to set this policy.

If the use of +/- syntax is desired in conjunction with **nisplus**, use the following four entries:

```
passwd:      compat
passwd_compat: nisplus
group:       compat
group_compat: nisplus
```

In order to get information from the Internet Domain Name Service for hosts that are not listed in the enterprise level name service, NIS+, use the following configuration and set up the `/etc/resolv.conf` file (see `resolv.conf(4)` for more details):

```
hosts:          nisplus dns [NOTFOUND=return] files
```

#### Enumeration -- getXXXent()

Many of the databases have enumeration functions: `passwd` has `getpwent()`, `hosts` has `gethostent()`, and so on. These were reasonable when the only source was `files` but often make little sense for hierarchically structured sources that contain large numbers of entries, much less for multiple sources. The interfaces are still provided and the implementations strive to provide reasonable results, but the data returned may be incomplete (enumeration for `hosts` is simply not supported by the `dns` source), inconsistent (if multiple sources are used), formatted in an unexpected fashion (for a host with a canonical name and three aliases, the `nisplus` source will return four hostents, and they may not be consecutive), or very expensive (enumerating a `passwd` database of 5,000 users is probably a bad idea). Furthermore, multiple threads in the same process using the same reentrant enumeration function (`getXXXent_r()`) are supported beginning with SunOS 5.3) share the same enumeration position; if they interleave calls, they will enumerate disjoint subsets of the same database.

In general, the use of the enumeration functions is deprecated. In the case of `passwd`, `shadow`, and `group`, it may sometimes be appropriate to use `fgetgrent()`, `fgetpwent()`, and `fgetspent()` (see `getgrnam(3C)`, `getpwnam(3C)`, and `getspnam(3C)`, respectively), which use only the `files` source.

#### FILES

A source named SSS is implemented by a shared object named `nss_SSS.so.1` that resides in `/usr/lib`.

<code>/etc/nsswitch.conf</code>	configuration file
<code>/usr/lib/nss_compat.so.1</code>	implements "compat" source
<code>/usr/lib/nss_dns.so.1</code>	implements "dns" source
<code>/usr/lib/nss_files.so.1</code>	implements "files" source
<code>/usr/lib/nss_nis.so.1</code>	implements "nis" source
<code>/usr/lib/nss_nisplus.so.1</code>	implements "nisplus" source
<code>/etc/netconfig</code>	configuration file for <code>netdir(3N)</code> functions that redirects hosts/devices policy to the switch
<code>/etc/nsswitch.files</code>	sample configuration file that uses "files" only
<code>/etc/nsswitch.nis</code>	sample configuration file that uses "files" and "nis"
<code>/etc/nsswitch.nisplus</code>	sample configuration file that uses "files" and "nisplus"

#### SEE ALSO

`nis+(1)`, `passwd(1)`, `automount(1M)`, `ifconfig(1M)`, `rpc.bootparamd(1M)`, `rpc.nisd(1M)`, `sendmail(1M)`, `ethers(3N)`, `getgrnam(3C)`, `gethostbyname(3N)`, `getnetbyname(3N)`, `getnetgrent(3N)`, `getprotobyname(3N)`, `getpublickey(3N)`, `getpwnam(3C)`, `getrpcbyname(3N)`, `getservbyname(3N)`, `getspnam(3C)`, `netdir(3N)`, `secure_rpc(3N)`, `netconfig(4)`, `resolv.conf(4)`, `ypfiles(4)`, `fns(5)`, `fns_policies(5)`

**NOTES**

Within each process that uses **nsswitch.conf**, the entire file is read only once; if the file is later changed, the process will continue using the old configuration.

Programs that use the **getXXbyYY()** functions cannot be linked statically since the implementation of these functions requires dynamic linker functionality to access the shared objects **/usr/lib/nss\_SSS.so.1** at run time.

The use of both **nis** and **nisplus** as sources for the same database is strongly discouraged since both the name services are expected to store similar information and the lookups on the database may yield different results depending on which name service is operational at the time of the request.

The **compat** source may not be supported in future releases.

Misspelled names of sources and databases will be treated as legitimate names of (most likely nonexistent) sources and databases.

The following functions do *not* use the switch: **fgetgrent(3C)**, **fgetpwent(3C)**, **fgetspent(3C)**, **getpw(3C)**, **putpwent(3C)**, **shadow(4)**.

<b>NAME</b>	order – package installation order description file
<b>DESCRIPTION</b>	<p>The package installation order file, <b>.order</b>, is an ASCII file specifying the order in which packages must be installed based on their prerequisite dependencies. Any package with prerequisite dependencies must be installed <i>after</i> any packages it lists as a prerequisite dependency in its <b>depend</b> file.</p> <p>A <b>.order</b> file is required for the OS product. The <b>.order</b> file must reside in the top-level directory containing the product.</p> <p>The ordering is specified as a list of package identifiers, from the first package to be installed to the last, one package identifier per line.</p>
<b>NOTES</b>	The <b>depend</b> file supports <i>incompatible</i> and <i>reverse</i> dependencies. These dependency types are not recognized in the <b>order</b> file.
<b>SEE ALSO</b>	<b>cdtoc(4)</b> , <b>clustertoc(4)</b> , <b>depend(4)</b> , <b>packagetoc(4)</b> , <b>pkginfo(4)</b>



<b>NAME</b>	ott – FACE object architecture information
<b>DESCRIPTION</b>	<p>The FACE object architecture stores information about object-types in an ASCII file named <b>.ott</b> (object type table) that is contained in each directory. This file describes all of the objects in that directory. Each line of the <b>.ott</b> file contains information about one object in pipe-separated fields. The fields are (in order):</p> <p><i>name</i>                    the name of the actual system file.</p> <p><i>dname</i>                    the name that should be displayed to the user, or a dot if it is the same as the name of the file.</p> <p><i>description</i>            the description of the object, or a dot if the description is the default (the same as object-type).</p> <p><i>object-type</i>            the FACE internal object type name.</p> <p><i>flags</i>                    object specific flags.</p> <p><i>mod time</i>                the time that FACE last modified the object. The time is given as number of seconds since 1/1/1970, and is in hexadecimal notation.</p> <p><i>object information</i>    an optional field, contains a set of semi-colon separated <i>name=value</i> fields that can be used by FACE to store any other information necessary to describe this object.</p>
<b>FILES</b>	<b>.ott</b> is created in any directory opened by FACE.

<b>NAME</b>	packagetoc – package table of contents description file
<b>DESCRIPTION</b>	<p>The package table of contents file, <b>.packagetoc</b>, is an ASCII file containing all of the information necessary for installing a product release distributed in package form. It centralizes and summarizes all of the relevant information about each package in the product. This allows the install software to quickly read one file to obtain all of the relevant information about each package instead of having to examine each package at run time to obtain this information. The <b>.packagetoc</b> file resides in the top-level directory containing the product.</p> <p>If a <b>.packagetoc</b> file exists for a product, there must also be a <b>.order</b> file.</p> <p>Each entry in the <b>.packagetoc</b> file is a line that establishes the value of a parameter in the following form:</p> <p style="text-align: center;"><b>PARAM=value</b></p> <p>A line starting with a pound-sign, “#”, is considered a comment and is ignored.</p> <p>Parameters are grouped by package. The start of a package description is defined by a line of the form:</p> <p style="text-align: center;"><b>PKG=value</b></p> <p>There is no order implied or assumed for specifying the parameters for a package with the exception of the <b>PKG</b> parameter, which must appear first. Only one occurrence of a parameter is permitted per package.</p> <p>The parameters recognized are described below. Those marked with an asterisk are mandatory.</p> <p><b>PKG*</b>                   The package identifier (for example, <b>SUNWaccu</b>). The maximum length of the identifier is nine characters. All the characters must be alphanumeric. The first character must be alphabetic. <b>install</b>, <b>new</b>, and <b>all</b> are reserved identifiers.</p> <p><b>PKGDIR*</b>               The name of the directory containing the package. This directory is relative to the directory containing the product.</p> <p><b>NAME*</b>                   The full name of the package.</p> <p><b>VENDOR</b>                The name of the package’s vendor.</p> <p><b>VERSION</b>               The version of the package.</p> <p><b>PRODNAME</b>             The name of the product to which this package belongs.</p> <p><b>PRODVERS</b>            The version of the product to which this package belongs.</p> <p><b>SUNW_PKGTYPE</b>        The package type. Valid values are:</p> <p style="padding-left: 2em;"><b>root</b>   indicates that the package will be installed in the / file system. The <b>root</b> packages are the only packages installed during dataless client installations. The <b>root</b> packages are spooled during a server installation to allow the later installation of diskless clients.</p> <p style="padding-left: 2em;"><b>usr</b>    indicates that the package will be installed in the /usr file</p>

	system.
	<b>kvm</b> indicates that the package will be installed in the <b>/usr/platform</b> file system.
	<b>ow</b> indicates a package that is part of the bundled OpenWindows product release. If no <b>SUNW_PKGTYPE</b> macro is present, the package is assumed to be of type <b>usr</b> .
<b>ARCH*</b>	The architecture(s) supported by the package. This macro is taken from the package's <b>pkginfo(4)</b> file and is subject to the same length and formatting constraints.  The install program currently assumes that exactly one architecture token is specified for a package. For example, <b>ARCH=sparc.sun4c</b> is acceptable, but <b>ARCH=sparc.sun4c, sparc.sun4m</b> is not.
<b>DESC</b>	A detailed textual description of the package.
<b>BASEDIR*</b>	The default installation base directory of the package.
<b>SUNW_PDEPEND</b>	A dependency specification for a prerequisite package. Each prerequisite dependency must appear as a separate macro. See <b>depend(4)</b> for more information on dependencies and instance specifications.
<b>SUNW_IDEPEND</b>	A dependency specification for an incompatible package. Each incompatible dependency should appear as a separate macro. See <b>depend(4)</b> for more information on dependencies and instance specifications.
<b>SUNW_RDEPEND</b>	A dependency specification for a reversed package dependency. Each reverse dependency should appear as a separate macro. See <b>depend(4)</b> for more information on dependencies and instance specifications.
<b>CATEGORY</b>	The category of the package.
<b>SUNW_LOC</b>	Indicates that this package contains localizations for other packages. Such localization packages are treated as special case packages. Each package which has a <b>SUNW_LOC</b> macro must have a corresponding <b>SUNW_PKGLIST</b> macro. The value specified by this macro should be a valid locale.
<b>SUNW_PKGLIST</b>	A comma separated list of package identifiers. Currently this macro is used to indicate which packages are localized by a localization package.
<b>ROOTSIZE*</b>	The space used by the package in the <b>/</b> file system.
<b>USRSIZE*</b>	The space used by the package in the <b>/usr</b> subtree of the file system.
<b>VARSIZE*</b>	The space used by the package in the <b>/var</b> subtree of the file system.
<b>OPTSIZE*</b>	The space used by the package in the <b>/opt</b> subtree of the file system.
<b>EXPORTSIZE*</b>	The space used by the package in the <b>/export</b> subtree of the file system.
<b>USROWNSIZE*</b>	The space used by the package in the <b>/usr/openwin</b> subtree of the file

system.

**SPOOLEDSIZE\*** The space used by the spooled version of this package. This is used during the setup of a server by the initial system installation programs.

All sizes are specified in bytes. Default disk partitions and file system sizes are derived from the values provided: accuracy is important.

#### EXAMPLES

The following is an example package entry in a **.packagetoc** file.

```
#ident "@(#)packagetoc.4 1.2 92/04/28"
PKG=SUNWaccr
PKGDIR=SUNWaccr
NAME=System Accounting, (Root)
VENDOR=Sun Microsystems, Inc.
VERSION=8.1
PRODNAME=SunOS
PRODVERS=5.0beta2
SUNW_PKGTYPE=root
ARCH=sparc
DESC=System Accounting, (Root)
BASEDIR=/
CATEGORY=system
ROOTSIZE=11264
VARSIZE= 15360
OPTSIZE=0
EXPORTSIZE=0
USRSIZE=0
USROWNSIZE=0
```

#### SEE ALSO

**cdtoc(4)**, **clustertoc(4)**, **depend(4)**, **order(4)**, **pkginfo(4)**, **pkgmap(4)**

#### NOTES

The parameters **NAME**, **VENDOR**, **VERSION**, **PRODNAME**, **PRODVERS**, **SUNW\_PKGTYPE**, **SUNW\_LOC**, **SUNW\_PKGLIST**, **ARCH**, **DESC**, **BASEDIR**, and **CATEGORY** are assumed to have been taken directly from the package's **pkginfo(4)** file. The length and formatting restrictions placed on the values for these parameters are identical to those for the corresponding entries in the **pkginfo(4)** file.

The value specified for the parameter **PKGDIR** should not exceed 255 characters.

The value specified for the parameters **ROOTSIZE**, **VARSIZE**, **OPTSIZE**, **EXPORTSIZE**, **USRSIZE** and **USROWNSIZE** must be a single integer value. The values can be derived from the package's **pkgmap** file by counting all space consumed by any files installed in the applicable file system. The space includes that used for directory entries and any UFS overhead that exists because of the way the files are represented (directory allocation scheme; direct, indirect, double indirect blocks; fragments; etc.)

The following kinds of entries in the **pkgmap(4)** file should be included in the space derivation:

- f** regular file
- c** character special file
- b** block special file
- p** pipe
- l** hard link
- s** symbolic link
- x, d** directory
- i** packaging installation script or information file (*copyright, depend, postinstall, postremove*)

<b>NAME</b>	packingrules – packing rules file for cachefs and filesync
<b>SYNOPSIS</b>	<b>\$HOME/.packingrules</b>
<b>DESCRIPTION</b>	<p><b>\$HOME/.packingrules</b> is a packing rules file for <b>filesync</b> and <b>cachefspack</b>. <b>\$HOME/.packingrules</b> contains a list of directories and files that are to be packed and synchronized. It also contains a list of directories and files that are to be specifically excluded from packing and synchronization. See <b>filesync(1)</b> and <b>cachefspack(1M)</b>.</p> <p>The <b>\$HOME/.packingrules</b> file is automatically created if users invoke <b>filesync</b> with filename arguments. By using <b>filesync</b> options, users can augment the packing rules in <b>\$HOME/.packingrules</b>.</p> <p>Many users choose to manually create the packing rules file and edit it by hand. Users can edit <b>\$HOME/.packingrules</b> (using any editor) to permanently change the <b>\$HOME/.packingrules</b> file, or to gain access to more powerful options that are not available from the command line (such as <b>IGNORE</b> commands). It is much easier to enter complex wildcard expressions by editing the <b>\$HOME/.packingrules</b> file.</p> <p>Blank lines and lines that begin with a pound sign (<b>#</b>) are ignored.</p> <p>Any line can be continued by placing a backslash (<b>\</b>) immediately before the NEWLINE. All other lines in the <b>\$HOME/.packingrules</b> file have one of the following formats:</p> <p style="margin-left: 40px;"><b>PACKINGRULES</b>                      <i>major. minor</i>. This line is not actually required, but it should be the first line of every packing rules file. This line identifies the packing rules file for the <b>file(1)</b> command and specifies a format version number. The current version number is 1.1. See <b>file(1)</b>.</p> <p style="margin-left: 40px;"><b>BASE</b> <i>directory-1</i> [<i>directory-2</i>] This line identifies a directory (or pair of directories) under which files should be packed and synchronized. At least one directory name must be specified. For rules that are to be used by <b>filesync</b> a second directory name (where the copies are to be kept) must also be specified. The arguments must be fully qualified path names, and may include environment variables.</p> <p style="margin-left: 40px;"><b>LIST</b> <i>name ...</i> This line enumerates a list of files and sub-directories (beneath the current <b>BASE</b>) that are to be kept synchronized. This specification is recursive, in that specifying the name of a directory automatically includes all files and subdirectories it contains. Regular expressions (as described in <b>glob</b> and <b>gmatch</b>) are permitted. See <b>glob(1)</b> and <b>gmatch(3)</b>.</p>

**IGNORE** *name* ...

This line enumerates a list of files that are not to be kept synchronized. Regular expressions (using **glob** and **gmatch**) are permitted.

There are important differences between the arguments to **LIST** and **IGNORE** statements. The arguments to a **LIST** statement can contain slashes and are interpreted as file names relative to the **BASE** directories. The arguments to an **IGNORE** statement are simpler names or expressions that cannot contain slashes. An **IGNORE** statement will not override a **LIST** statement. **IGNORE** statements only exclude files that are found beneath **LIST**ed directories.

If the first name argument to a **LIST** statement begins with an exclamation point ('!'), the remainder of the statement will be executed as a command. The command will be run in the current **BASE** directory. The output of the command will be treated as a list of newline separated file names to be packed/synchronized. The resulting file names will be interpreted relative to the enclosing **BASE** directory.

If the first name argument to an **IGNORE** statement begins with an exclamation point ('!'), the remainder of the statement will be executed as a command. The command will be run in the current **BASE** directory. The command will be expected to figure out which names should not be synchronized. The output of the command will be treated as a list of newline separated file names that should be excluded from the packing and synchronization list.

Commands will be broken into distinct arguments and run directly with **sh -c**. Blanks can be embedded in an argument by escaping them with a backslash ('\') or enclosing the argument in double quotes (' '). Double quotes can be passed in arguments by escaping the double quotes with a backslash ('\').

**LIST** lines only apply to the **BASE** statement that precedes them. **IGNORE** lines can appear before any **BASE** statement (in which case they apply to all **BASE**s) or after a **BASE** statement (in which case they only apply to the **BASE** that precedes them). Any number of these statements can occur in any combination. The order is not important.

## EXAMPLES

The use of these statements is illustrated in the following **\$HOME.packingrules** file.

```
#
# junk files, not worth copying
#
IGNORE core *.o *.bak *%

#
# most of the stuff I want to keep in sync is in my $HOME
#
BASE /net/bigserver/export/home/myname $HOME

# everything in my work sub-directory should be maintained
LIST work
```

```
# a few of my favorite mail boxes should be replicated  
LIST m/incoming  
LIST m/action  
LIST m/pending  
  
#  
# I like to carry around a couple of project directories  
# but skip all the postscript output  
#  
BASE /net/bigserver/export/projects $HOME/projects  
LIST poindexter epiphany  
IGNORE *.ps  
  
#  
# the foonly package should always be kept on every machine  
#  
BASE /net/bigserver/opt/foonly /opt/foonly  
LIST !cat .packinglist  
  
#  
# and the latest executables for the standard build environment  
#  
BASE /net/bigserver/export/buildenv $HOME/buildenv  
LIST !find . -type f -a -perm -111 -a -print
```

**SEE ALSO** file(1), filesync(1), cacheftp(1M)



<b>NAME</b>	pam.conf – configuration file for pluggable authentication modules
<b>SYNOPSIS</b>	<i>/etc/pam.conf</i>
<b>DESCRIPTION</b>	<p><b>pam.conf</b> is the configuration file for the Pluggable Authentication Module architecture, or PAM. A PAM module provides functionality for one or more of four possible services: authentication, account management, session management, and password management. An authentication service module provides functionality to authenticate a user and set up user credentials. A account management module provides functionality to determine if the current user's account is valid. This includes checking for password and account expiration, as well as verifying access hour restrictions. A session management module provides functionality to set up and terminate login sessions. A password management module provides functionality to change a user's authentication token or password. Each of the four service modules can be implemented as a shared library object which can be referenced in the <b>pam.conf</b> configuration file.</p> <p>The <b>pam.conf</b> file contains a listing of services. Each service is paired with a corresponding service module. When a service is requested, its associated module is invoked. Each entry has the following format:</p> <pre style="margin-left: 40px;"><i>&lt;service_name&gt; &lt;module_type&gt; &lt;control_flag&gt; &lt;module_path&gt; &lt;options&gt;</i></pre> <p>Below is an example of the <b>pam.conf</b> configuration file with support for authentication, account management, and session management modules.</p> <pre style="margin-left: 40px;"> <b>login</b>   <b>auth</b>       <b>required</b>  <i>/usr/lib/security/pam_unix.so.1</i>  <b>debug</b> <b>login</b>   <b>session</b>    <b>required</b>  <i>/usr/lib/security/pam_unix.so.1</i> <b>login</b>   <b>account</b>   <b>required</b>  <i>/usr/lib/security/pam_unix.so.1</i> <b>telnet</b>  <b>session</b>    <b>required</b>  <i>/usr/lib/security/pam_unix.so.1</i> <b>other</b>   <b>auth</b>       <b>required</b>  <i>/usr/lib/security/pam_unix.so.1</i> <b>other</b>   <b>passwd</b>    <b>required</b>  <i>/usr/lib/security/pam_unix.so.1</i></pre> <p>The <i>service_name</i> denotes the service (for example, <b>login</b>, <b>dtlogin</b>, or <b>rlogin</b>). The keyword, <i>other</i>, indicates the module all other applications which have not been specified should use. The <i>other</i> keyword can also be used if all services of the same <i>module_type</i> have the same requirements. In the example above, since all of the services use the same session module, they could have been replaced by a single <i>other</i> line.</p> <p><i>module_type</i> denotes the service module type: authentication (<i>auth</i>), account management (<i>account</i>), session management (<i>session</i>), or password management (<i>password</i>).</p> <p>The <i>control_flag</i> field determines the behavior of stacking, and will be discussed in more detail below.</p> <p>The <i>module_path</i> field specifies the pathname to a shared library object which implements the service functionality. If the pathname is not absolute, it is assumed to be relative to <i>/usr/lib/security</i>.</p>
<b>Simplified PAM.CONF configuration file</b>	

### Integrating Multiple Authentication Services With Stacking

The *options* field is used by the PAM framework layer to pass module specific options to the modules. It is up to the module to parse and interpret the options. This field can be used by the modules to turn on debugging or to pass any module specific parameters such as a TIMEOUT value. It can also be used to support unified login. The options supported by the modules are documented in their respective manual pages. For example, **pam\_unix**(5) lists the options accepted by the UNIX module.

When a service\_name of the same *module\_type* is defined more than once, the service is said to be *stacked*. Each module referenced in the *module\_path* for that service is then processed in the order that it occurs in the configuration file. The *control\_flag* field specifies the continuation and failure semantics of the modules, and may be *requisite*, *required*, *optional*, or *sufficient*.

The PAM framework processes each service module in the stack. If all *requisite* and *required* modules in the stack succeed, then success is returned, and *optional* and *sufficient* error values are ignored. If one or more *requisite* or *required* modules fail, then the error value from the first *requisite* or *required* module that failed is returned.

If none of the service modules in the stack are designated as *requisite* or *required*, then the PAM framework requires that at least one *optional* or *sufficient* module succeed. If all fail then the error value from the first service module in the stack is returned.

The *requisite* and *sufficient* flags cause two exceptions to the above semantics. If a service module that is designated as *requisite* fails, then the PAM framework immediately returns an error to the application, and all subsequent service modules in the stack are ignored. If a prior *required* service module has failed, then that error is returned. If no prior *required* service module failed, then the error from the failed *requisite* service module is returned.

If a service module that is designated as *sufficient* succeeds, then the PAM framework immediately returns success to the application, and all subsequent services modules in the stack, even *requisite* and *required* ones, are ignored, given that all prior *requisite* and *required* modules have also succeeded. If a prior *required* module has failed, then the error value from that module is returned.

If any entry in **pam.conf** is incorrect, or if a module does not exist or cannot be opened, then all PAM services will fail and users will not be permitted access to the system. An error will be logged through **syslog**(3) at the LOG\_CRIT level. To fix incorrect entries in **pam.conf**, a system administrator may boot the system in maintenance mode (single user) to edit the file. Below is a sample configuration file that stacks the **su**, **login**, and **rlogin** services.

```
su      auth  requisite  /usr/lib/security/pam_inhouse.so.1
su      auth  required   /usr/lib/security/pam_unix.so.1      debug
login   auth  required   /usr/lib/security/pam_unix.so.1      debug
login   auth  optional   /usr/lib/security/pam_inhouse.so.1
rlogin  auth  sufficient /usr/lib/security/pam_rhosts_auth.so.1
rlogin  auth  required   /usr/lib/security/pam_unix.so.1
```

In the case of **su**, the user is authenticated by the Inhouse and UNIX authentication modules. Because the Inhouse and UNIX authentication modules are *requisite* and *required*, respectively, an error is returned back to the application if either module fails. In addition, if the *requisite* authentication (Inhouse authentication) fails, the UNIX authentication module is never invoked, and the error is returned immediately back to the application.

In the case of **login**, the *required* keyword for *control\_flag* requires that the user be allowed to login only if the user is authenticated by the UNIX service module. If UNIX authentication fails, control continues to proceed down the stack, and the Inhouse authentication module is invoked. Inhouse authentication is optional by virtue of the *optional* keyword in the *control\_flag* field. The user can still log in even if Inhouse authentication fails, assuming the UNIX authentication succeeded.

In the case of **rlogin**, the *sufficient* keyword for *control\_flag* specifies that if the *rhosts* authentication check succeeds, then PAM should return success to **rlogin** and **rlogin** should not prompt the user for a password. The UNIX authentication module, which is the next module in the stack, will only be invoked if the *rhosts* check fails. This gives the system administrator the flexibility to determine if *rhosts* alone is sufficient enough to authenticate a remote user.

Some modules may return PAM\_IGNORE in certain situations. In these cases the PAM framework ignores the entire entry in **pam.conf** regardless of whether or not it is *requisite*, *required*, *optional* or *sufficient*.

#### Utilities and Files

A following is a list of the utilities that are known to use PAM: include: **login**, **passwd**, **su**, **rlogind**, **rshd**, **telnetd**, **ftpd**, **rpc.rexd**, **uucpd**, **init**, **sac**, and **ttymon**.

The utility **dtlogin** also uses PAM. Note however that **dtlogin** is the login service utility for the Common Desktop Environment (CDE).

The PAM configuration file does not dictate either the name or the location of the service specific modules. The convention, however, is the following:

```
/usr/lib/security/pam_<module_name>.so.x    Implements various function of specific authentication services.
/etc/pam.conf                               Configuration file.
/usr/lib/libpam.so.1                       Implements the PAM framework library.
```

#### EXAMPLES

The following is a sample pam.conf configuration file. Lines that begin with the # symbol are treated as comments, and therefore ignored.

```
#
# PAM configuration
#
# Authentication management for login service is stacked.
# Both UNIX and inhouse authentication functions are invoked.
login    auth    required    /usr/lib/security/pam_unix.so.1
login    auth    required    /usr/lib/security/pam_inhouse.so.1  try_first_pass
dtlogin  auth    required    /usr/lib/security/pam_unix.so.1
```

```

dtlogin  auth  required  /usr/lib/security/pam_inhouse.so.1  try_first_pass
#
# Authentication management for rlogin service is stacked.
# If the rhost check succeeds, do not continue
rlogin   auth  sufficient /usr/lib/security/pam_rhosts_auth.so.1
rlogin   auth  required  /usr/lib/security/pam_unix.so.1
#
# Other services use UNIX authentication
other    auth  required  /usr/lib/security/pam_unix.so.1
#
# Account management for login service is stacked.
# UNIX account management is required
# Inhouse account management is optional
login    account  required  /usr/lib/security/pam_unix.so.1
login    account  optional  /usr/lib/security/pam_inhouse.so.1
dtlogin  account  required  /usr/lib/security/pam_unix.so.1
dtlogin  account  optional  /usr/lib/security/pam_inhouse.so.1
other    account  required  /usr/lib/security/pam_unix.so.1
#
# Session management
other    session  required  /usr/lib/security/pam_unix.so.1
#
# Password management
other    password  required  /usr/lib/security/pam_unix.so.1

```

**ATTRIBUTES**

See [attributes\(5\)](#) for description of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT Level	MT-Safe with exceptions

**SEE ALSO**

[login\(1\)](#), [passwd\(1\)](#), [in.ftpd\(1M\)](#), [in.rlogind\(1M\)](#), [in.rshd\(1M\)](#), [in.telnetd\(1M\)](#), [in.uucpd\(1M\)](#), [init\(1M\)](#), [rpc.rexd\(1M\)](#), [sac\(1M\)](#), [su\(1M\)](#), [ttypmon\(1M\)](#), [pam\(3\)](#), [syslog\(3\)](#), [libpam\(4\)](#), [attributes\(5\)](#), [pam\\_unix\(5\)](#)

**NOTES**

The interfaces in **libpam()** are MT-Safe only if each thread within the multi-threaded application uses its own PAM handle.

<b>NAME</b>	passwd – password file														
<b>SYNOPSIS</b>	<code>/etc/passwd</code>														
<b>DESCRIPTION</b>	<p><code>/etc/passwd</code> is a local source of information about users' accounts. The password file can be used in conjunction with other password sources, including the NIS maps <code>passwd.byname</code> and <code>passwd.bygid</code> and the NIS+ table <code>passwd</code>. Programs use the <code>getpwnam(3C)</code> routines to access this information.</p> <p>Each <code>passwd</code> entry is a single line of the form:</p> <pre>username:password:uid:gid:gcoss-field:home-dir:login-shell</pre> <p>where</p> <table border="0"> <tr> <td style="padding-right: 1em;"><i>username</i></td> <td>is the user's login name. It is recommended that this field conform to the checks performed by <code>pwck(1M)</code>.</td> </tr> <tr> <td><i>password</i></td> <td>is an empty field. The encrypted password for the user is in the corresponding entry in the <code>/etc/shadow</code> file. <code>pwconv(1M)</code> relies on a special value of 'x' in the password field of <code>/etc/passwd</code>. If this value of 'x' exists in the password field of <code>/etc/passwd</code>, this indicates that the password for the user is already in <code>/etc/shadow</code> and should not be modified.</td> </tr> <tr> <td><i>uid</i></td> <td>is the user's unique numerical ID for the system.</td> </tr> <tr> <td><i>gid</i></td> <td>is the unique numerical ID of the group that the user belongs to.</td> </tr> <tr> <td><i>gcoss-field</i></td> <td>is the user's real name, along with information to pass along in a mail-message heading. (It is called the gcoss-field for historical reasons.) An "&amp;" (ampersand) in this field stands for the login name (in cases where the login name appears in a user's real name).</td> </tr> <tr> <td><i>home-dir</i></td> <td>is the pathname to the directory in which the user is initially positioned upon logging in.</td> </tr> <tr> <td><i>login-shell</i></td> <td>is the user's initial shell program. If this field is empty, the default shell is <code>/usr/bin/sh</code>.</td> </tr> </table> <p>The maximum value of the <i>uid</i> and <i>gid</i> fields is <b>2147483647</b>. To maximize interoperability and compatibility, administrators are recommended to assign users a range of UIDs and GIDs below <b>60000</b> where possible.</p> <p>The password file is an ASCII file. Because the encrypted passwords are always kept in the shadow file, <code>/etc/passwd</code> has general read permission on all systems and can be used by routines that map between numerical user IDs and user names.</p> <p>Previous releases used a password entry beginning with a '+' (plus sign) or '-' (minus sign) to selectively incorporate entries from NIS maps for password. If still required, this is supported by specifying "passwd : compat" in <code>nsswitch.conf(4)</code>. The "compat" source may not be supported in future releases. The preferred sources are, "files" followed by "nisplus". This has the effect of incorporating the entire contents of the NIS+ <code>passwd</code></p>	<i>username</i>	is the user's login name. It is recommended that this field conform to the checks performed by <code>pwck(1M)</code> .	<i>password</i>	is an empty field. The encrypted password for the user is in the corresponding entry in the <code>/etc/shadow</code> file. <code>pwconv(1M)</code> relies on a special value of 'x' in the password field of <code>/etc/passwd</code> . If this value of 'x' exists in the password field of <code>/etc/passwd</code> , this indicates that the password for the user is already in <code>/etc/shadow</code> and should not be modified.	<i>uid</i>	is the user's unique numerical ID for the system.	<i>gid</i>	is the unique numerical ID of the group that the user belongs to.	<i>gcoss-field</i>	is the user's real name, along with information to pass along in a mail-message heading. (It is called the gcoss-field for historical reasons.) An "&" (ampersand) in this field stands for the login name (in cases where the login name appears in a user's real name).	<i>home-dir</i>	is the pathname to the directory in which the user is initially positioned upon logging in.	<i>login-shell</i>	is the user's initial shell program. If this field is empty, the default shell is <code>/usr/bin/sh</code> .
<i>username</i>	is the user's login name. It is recommended that this field conform to the checks performed by <code>pwck(1M)</code> .														
<i>password</i>	is an empty field. The encrypted password for the user is in the corresponding entry in the <code>/etc/shadow</code> file. <code>pwconv(1M)</code> relies on a special value of 'x' in the password field of <code>/etc/passwd</code> . If this value of 'x' exists in the password field of <code>/etc/passwd</code> , this indicates that the password for the user is already in <code>/etc/shadow</code> and should not be modified.														
<i>uid</i>	is the user's unique numerical ID for the system.														
<i>gid</i>	is the unique numerical ID of the group that the user belongs to.														
<i>gcoss-field</i>	is the user's real name, along with information to pass along in a mail-message heading. (It is called the gcoss-field for historical reasons.) An "&" (ampersand) in this field stands for the login name (in cases where the login name appears in a user's real name).														
<i>home-dir</i>	is the pathname to the directory in which the user is initially positioned upon logging in.														
<i>login-shell</i>	is the user's initial shell program. If this field is empty, the default shell is <code>/usr/bin/sh</code> .														

table after the password file.

## EXAMPLES

Here is a sample **passwd** file:

```
root:q.mJzTnu8icF.:0:10:God:~/bin/csh
fred:6k/7KCFRPNVXg:508:10:% Fredericks:/usr2/fred:/bin/csh
```

and the sample password entry from **nsswitch.conf**:

```
passwd: files nisplus
```

In this example, there are specific entries for users **root** and **fred** to assure that they can login even when the system is running single-user. In addition, anyone in the NIS+ table **passwd** will be able to login with their usual password, shell and home directory.

If the password file is:

```
root:q.mJzTnu8icF.:0:10:God:~/bin/csh
fred:6k/7KCFRPNVXg:508:10:% Fredericks:/usr2/fred:/bin/csh
+
```

and the password entry from **nsswitch.conf** is:

```
passwd: compat
```

then all the entries listed in the NIS **passwd.byuid** and **passwd.byname** maps will be effectively incorporated after the entries for **root** and **fred**.

## FILES

```
/etc/nsswitch.conf
/etc/passwd
/etc/shadow
```

## SEE ALSO

**chgrp**(1), **chown**(1), **groups**(1), **login**(1), **makekey**(1), **newgrp**(1), **nispasswd**(1), **passwd**(1), **sh**(1), **sort**(1), **chown**(1M), **domainname**(1M), **getent**(1M), **in.ftpd**(1M), **passmgmt**(1M), **pwck**(1M), **pwconv**(1M), **su**(1M), **useradd**(1M), **userdel**(1M), **usermod**(1M), **a64l**(3C), **crypt**(3C), **getpw**(3C), **getpwnam**(3C), **getspnam**(3C), **putpwent**(3C), **group**(4), **hosts.equiv**(4), **nsswitch.conf**(4), **shadow**(4), **unistd**(4), **environ**(5)

*System Administration Guide*

<b>NAME</b>	pathalias – alias file for FACE
<b>SYNOPSIS</b>	<code>/usr/vmsys/pathalias</code>
<b>DESCRIPTION</b>	<p>The <b>pathalias</b> files contain lines of the form <b>alias=path</b> where <i>path</i> can be one or more colon-separated directories. Whenever a FACE (Framed Access Command Environment, see <b>face(1)</b>) user references a path not beginning with a “/”, this file is checked. If the first component of the pathname matches the left-hand side of the equals sign, the right-hand side is searched much like <b>\$PATH</b> variable in the system. This allows users to reference the folder <b>\$HOME/FILECABINET</b> by typing <b>filecabinet</b>.</p> <p>There is a system-wide <b>pathalias</b> file called <b>\$VMSYS/pathalias</b>, and each user can also have local alias file called <b>\$HOME/pref/pathalias</b>. Settings in the user alias file override settings in the system-wide file. The system-wide file is shipped with several standard FACE aliases, such as <b>filecabinet</b>, <b>wastebasket</b>, <b>preferences</b>, <b>other_users</b>, etc.</p>
<b>FILES</b>	<b>\$HOME/pref/pathalias</b> <b>\$VMSYS/pathalias</b>
<b>SEE ALSO</b>	<b>face(1)</b>
<b>NOTES</b>	Unlike command keywords, partial matching of a path alias is not permitted, however, path aliases are case insensitive. The name of an alias should be alphabetic, and in no case can it contain special characters like “/”, “\”, or “=”. There is no particular limit on the number of aliases allowed. Alias files are read once, at login, and are held in core until logout. Thus, if an alias file is modified during a session, the change will not take effect until the next session.

<b>NAME</b>	path_to_inst – device instance number file						
<b>SYNOPSIS</b>	<code>/etc/path_to_inst</code>						
<b>DESCRIPTION</b>	<p><code>/etc/path_to_inst</code> records mappings of physical device names to instance numbers. The instance number of a device is encoded in its minor number, and is the way that a device driver determines which of the possible devices that it may drive is referred to by a given special file.</p> <p>In order to keep instance numbers persistent across reboots, the system records them in <code>/etc/path_to_inst</code>.</p> <p>This file is read only at boot time, and is updated by <code>add_drv(1M)</code> and <code>drvconfig(1M)</code>. Note that it is generally not necessary for the system administrator to change this file, as the system will maintain it.</p> <p>The system administrator can change the assignment of instance numbers by editing this file and doing a reconfiguration reboot. However, any changes made in this file will be lost if <code>add_drv(1M)</code> or <code>drvconfig(1M)</code> is run before the system is rebooted.</p> <p>Each instance entry is a single line of the form:</p> <pre style="margin-left: 40px;"><i>"physical name" instance number "driver binding name"</i></pre> <p>where</p> <table border="0" style="margin-left: 40px;"> <tr> <td><i>physical name</i></td> <td>is the absolute physical pathname of a device. This pathname must be enclosed in double quotes.</td> </tr> <tr> <td><i>instance number</i></td> <td>is a decimal or hexadecimal number.</td> </tr> <tr> <td><i>driver binding name</i></td> <td>is the name used to determine the driver for the device. This name may be a driver alias or a driver name. The driver binding name must be enclosed in double quotes.</td> </tr> </table>	<i>physical name</i>	is the absolute physical pathname of a device. This pathname must be enclosed in double quotes.	<i>instance number</i>	is a decimal or hexadecimal number.	<i>driver binding name</i>	is the name used to determine the driver for the device. This name may be a driver alias or a driver name. The driver binding name must be enclosed in double quotes.
<i>physical name</i>	is the absolute physical pathname of a device. This pathname must be enclosed in double quotes.						
<i>instance number</i>	is a decimal or hexadecimal number.						
<i>driver binding name</i>	is the name used to determine the driver for the device. This name may be a driver alias or a driver name. The driver binding name must be enclosed in double quotes.						
<b>EXAMPLES</b>	<p>Here are some sample <code>path_to_inst</code> entries:</p> <pre style="margin-left: 40px;">"/iommu@f,e0000000" 0 "iommu" "/iommu@f,e0000000/sbus@f,e0001000" 0 "sbus" "/iommu@f,e0000000/sbus@f,e0001000/sbusmem@e,0" 14 "sbusmem" "/iommu@f,e0000000/sbus@f,e0001000/sbusmem@f,0" 15 "sbusmem" "/iommu@f,e0000000/sbus@f,e0001000/ledma@f,400010" 0 "ledma" "/obio/serial@0,100000" 0 "zs" "/SUNW,sx@f,80000000" 0 "SUNW,sx"</pre>						
<b>FILES</b>	<code>/etc/path_to_inst</code>						
<b>SEE ALSO</b>	<code>add_drv(1M)</code> , <code>boot(1M)</code> , <code>drvconfig(1M)</code> , <code>mknod(1M)</code>						
<b>WARNINGS</b>	If the file is removed the system may not be bootable (as it may rely on information found in this file to find the root, usr or swap device). If it does successfully boot, it will regenerate the file, but after rebooting devices may end up having different minor numbers						



than they did before, and special files created via **mknod(1M)** may refer to different devices than expected.

For the same reasons, changes should not be made to this file without careful consideration.

**NOTES**

This document does not constitute an API. **path\_to\_inst** may not exist or may have a different content or interpretation in a future release. The existence of this notice does not imply that any other documentation that lacks this notice constitutes an API.

<b>NAME</b>	pci – configuration files for PCI device drivers
<b>DESCRIPTION</b>	<p>The Peripheral Component Interconnect (PCI) bus is a little endian bus. PCI devices are <i>self-identifying</i> — that is to say the PCI device provides configuration parameters to the system which allows the system to identify the device and its driver. The configuration parameters are represented in the form of name-value pairs that can be retrieved using the DDI property interfaces. See <b>ddi_prop_lookup(9F)</b> for details.</p> <p>The PCI bus properties are derived from PCI Configuration Space, or supplied by the Fcode PROM if it exists. Therefore, driver configuration files are not necessary for these devices.</p> <p>However, on some occasions, drivers for PCI devices may use driver configuration files to provide driver private properties. This can be done through global property mechanism. See <b>driver.conf(4)</b> for further details. Driver configuration files can also be used to augment or override properties for a specific instance of a driver.</p> <p>All bus drivers of class <b>pci</b> recognize the following properties:</p> <p><b>reg</b>           An arbitrary length array where each element of the array consists of a 5-tuple of 32-bit values. Each array element describes a logically contiguous mappable resource on the PCI bus.</p> <p>The first 3 values in the 5-tuple describe the PCI address of the mappable resource. The first tuple contains the following information:</p> <p><b>Bits 0 - 7       8-bit Register number</b>  <b>Bits 8 - 10     3-bit Function number</b>  <b>Bits 11 - 15    5-bit Device number</b>  <b>Bits 16 - 23    8-bit Bus number</b>  <b>Bits 24 - 25    2-bit Address Space type identifier</b></p> <p>The Address Space type identifier may be interpreted as follows:</p> <p><b>0x0   Configuration Space</b>  <b>0x1   I/O Space</b>  <b>0x2   32-bit Memory Space address</b>  <b>0x3   64-bit Memory Space address</b></p> <p>The Bus number is a unique identifying number assigned to each PCI bus within a PCI domain.</p> <p>The Device number is a unique identifying number assigned to each PCI device on a PCI bus. Note that a Device number is only unique within the set of Device numbers for a particular bus.</p> <p>Each PCI device can have 1 to 8 logically independent functions, each with its own independent set of configuration registers. Each function on a device is assigned a Function number. For a PCI device with only one function, the Function number must be 0.</p> <p>The Register number field selects a particular register within the set of configuration registers corresponding to the selected function.</p>

The second and third values in the **reg** property 5-tuple specify the 64-bit address of the mappable resource within the PCI address domain. The second 32-bit tuple corresponds to the high order 4 bytes of the 64-bit address. The third 32-bit tuple corresponds to the low order bytes.

The fourth and fifth 32-bit values in the 5-tuple **reg** property specify the size of the mappable resource. The size is a 64-bit value where the fourth tuple corresponds to the high order bytes of the 64-bit size and the fifth corresponds to the low order.

The driver can refer to the elements of this array by index, and construct kernel mappings to these addresses using **ddi\_regs\_map\_setup**(9F). The index into the array is passed as the *rnumber* argument of **ddi\_regs\_map\_setup**(9F).

### **interrupts**

This property consists of a single integer element array. Valid interrupt property values are **1**, **2**, **3**, and **4**. This value is derived directly from the contents of the device's Configuration Interrupt Pin register.

A driver should use an index value of **0** when registering its interrupt handler with **ddi\_add\_intr**(9F).

All PCI devices support the **reg** property. The Device number and Function number as derived from the **reg** property are used to construct the address part of the device name under **/devices**.

Only devices that generate interrupts support an **interrupts** property.

Occasionally it may be necessary to override or augment the configuration information supplied by a PCI device. This can be achieved by writing a driver configuration file that describes a prototype device node specification containing the additional properties required.

For the system to merge the prototype node specification into an actual device node, certain conditions must be met. First, the name property must be identical. Second, the parent property must identify the PCI bus. Third, the unit-address property must identify the card. The format of the unit-address property is

**DD[F]**

where DD is the device number and F is the function number. If the function number is 0, only DD is specified.

### **EXAMPLES**

An example configuration file called **ACME,scsi-hba.conf** for a PCI driver called **ACME,scsi-hba** follows:

```
#
# Copyright (c) 1995, ACME SCSI Host Bus Adaptor
# ident "@(#)ACME,scsi-hba.conf 1.1 96/02/04"
name="ACME,scsi-hba" parent="/pci@1,0/pci@1f,4000"
  unit-address="3" scsi-initiator-id=6;
hba-advanced-mode="on";
```

**hba-dma-speed=10;**

In this example, we provide a property `scsi-initiator-id` to specify the SCSI bus initiator id that the adapter should use, for just one particular instance of adapter installed in the machine. We use the `name` property to identify the driver and the `parent` property to identify the particular bus the card is plugged into. This example uses the parent's full path name to identify the bus. The `unit-address` property identifies the card itself, with device number of 3 and function number of 0.

Two global driver properties are also created: **hba-advanced-mode** (which has the string value **on**) and **hba-dma-speed** (which has the value **10 M bit/s**). These properties apply to all device nodes of the **ACME,scsi-hba**. The following is an example configuration file called **ACME,foo.conf** for a PCI driver called **ACME,foo**;

```
#
# Copyright (c) 1996, ACME Foo driver
# ident "@(#)ACME,foo.conf 1.1 95/11/14"
name="ACME,foo" class="pci" unit-address="3,1"
debug-mode=12;
```

In this example, we provide a property **debug-mode** for all instances of the **ACME,foo** driver with parents of class **pci** and device and function numbers of **3** and **1**, respectively.

**ATTRIBUTES**

See **attributes(5)** for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Architecture	SPARC, x86

**SEE ALSO**

**driver.conf(4)**, **attributes(5)**, **ddi\_add\_intr(9F)**, **ddi\_prop\_lookup(9F)**, **ddi\_regs\_map\_setup(9F)**

*Writing Device Drivers*

*IEEE 1275 PCI Bus Binding*

<b>NAME</b>	pcmcia – PCMCIA nexus driver
<b>DESCRIPTION</b>	The PCMCIA nexus driver supports PCMCIA card client device drivers. There are no user-configurable options for this driver.
<b>FILES</b>	<code>/kernel/misc/pcmcia</code> <code>pcmcia</code> driver
<b>SEE ALSO</b>	<code>pcmciad(1M)</code>

<b>NAME</b>	phones – remote host phone number database
<b>SYNOPSIS</b>	<b>/etc/phones</b>
<b>DESCRIPTION</b>	<p>The file <b>/etc/phones</b> contains the system-wide private phone numbers for the <b>tip(1)</b> program. <b>/etc/phones</b> is normally unreadable, and so may contain privileged information. The format of <b>/etc/phones</b> is a series of lines of the form:</p> <p style="text-align: center;"><i>&lt;system-name&gt;[ \t]*&lt;phone-number&gt;.</i></p> <p>The system name is one of those defined in the <b>remote(4)</b> file and the phone number is constructed from <b>[0123456789-=*%]</b>. The '=' and '*' characters are indicators to the auto call units to pause and wait for a second dial tone (when going through an exchange). The '=' is required by the DF02-AC and the '*' is required by the BIZCOMP 1030.</p> <p>Comment lines are lines containing a '#' sign in the first column of the line.</p> <p>Only one phone number per line is permitted. However, if more than one line in the file contains the same system name <b>tip(1)</b> will attempt to dial each one in turn, until it establishes a connection.</p>
<b>FILES</b>	<b>/etc/phones</b>
<b>SEE ALSO</b>	<b>tip(1), remote(4)</b>

<b>NAME</b>	pkginfo – package characteristics file
<b>DESCRIPTION</b>	<p><b>pkginfo</b> is an ASCII file that describes the characteristics of the package along with information that helps control the flow of installation. It is created by the software package developer.</p> <p>Each entry in the <b>pkginfo</b> file is a line that establishes the value of a parameter in the following form:</p> <p style="text-align: center;"><i>PARAM="value"</i></p> <p>There is no required order in which the parameters must be specified within the file. Each parameter is described below. Only fields marked with an asterisk are mandatory.</p> <p><b>PKG*</b> Abbreviation for the package being installed. All characters in the abbreviation must be alphanumeric and the first may not be numeric. The abbreviation is limited to a maximum length of nine characters. <b>install</b>, <b>new</b>, and <b>all</b> are reserved abbreviations. It is customary to make the first four letters unique to your company, such as the company's stock symbol.</p> <p><b>NAME*</b> Text that specifies the package name (maximum length of 256 ASCII characters).</p> <p><b>ARCH*</b> A comma-separated list of alphanumeric tokens that indicate the architecture associated with the package. The <b>pkgmk(1)</b> tool may be used to create or modify this value when actually building the package. The maximum length of a token is 16 characters and it cannot include a comma.</p> <p><b>VERSION*</b> Text that specifies the current version associated with the software package. The maximum length is 256 ASCII characters and the first character cannot be a left parenthesis. The <b>pkgmk(1)</b> tool may be used to create or modify this value when actually building the package.</p> <p><b>CATEGORY*</b> A comma-separated list of categories under which a package may be displayed. A package must at least belong to the system or application category. Categories are case-insensitive and may contain only alphanumerics. Each category is limited in length to 16 characters.</p> <p><b>DESC</b> Text that describes the package (maximum length of 256 ASCII characters).</p> <p><b>VENDOR</b> Used to identify the vendor that holds the software copyright (maximum length of 256 ASCII characters).</p> <p><b>HOTLINE</b> Phone number and/or mailing address where further information may be received or bugs may be reported (maximum length of 256 ASCII characters).</p>

<b>EMAIL</b>	An electronic address where further information is available or bugs may be reported (maximum length of 256 ASCII characters).
<b>VSTOCK</b>	The vendor stock number, if any, that identifies this product (maximum length of 256 ASCII characters).
<b>CLASSES</b>	A space-separated list of classes defined for a package. The order of the list determines the order in which the classes are installed. Classes listed first will be installed first (on a media by media basis). This parameter may be modified by the request script.
<b>ISTATES</b>	A list of allowable run states for package installation (for example, "S s 1").
<b>RSTATES</b>	A list of allowable run states for package removal (for example, "S s 1").
<b>BASEDIR</b>	The pathname to a default directory where "relocatable" files may be installed. If blank, the package is not relocatable and any files that have relative pathnames will not be installed. An administrator can override the default directory.
<b>ULIMIT</b>	If set, this parameter is passed as an argument to the <b>ulimit(1)</b> command (see <b>limit(1)</b> ), which establishes the maximum size of a file during installation.
<b>ORDER</b>	A list of classes defining the order in which they should be put on the medium. Used by <b>pkgmk(1)</b> in creating the package. Classes not defined in this field are placed on the medium using the standard ordering procedures.
<b>MAXINST</b>	The maximum number of package instances that should be allowed on a machine at the same time. By default, only one instance of a package is allowed. This parameter must be set in order to have multiple instances of a package.
<b>PSTAMP</b>	Production stamp used to mark the <b>pkgmap(4)</b> file on the output volumes. Provides a means for distinguishing between production copies of a version if more than one is in use at a time. If <b>PSTAMP</b> is not defined, the default is used. The default consists of the UNIX system machine name followed by the string "YYMMDDHHMM" (year, month, date, hour, minutes).
<b>INTONLY</b>	Indicates that the package should only be installed interactively when set to any non-NULL value.



**EXAMPLES**

Here is a sample **pkginfo** file:

```
PKG="oam"  
NAME="OAM Installation Utilities"  
VERSION="3"  
VENDOR="AT&T"  
HOTLINE="1-800-ATT-BUGS"  
EMAIL="attunix!olsen"  
VSTOCK="0122c3f5566"  
CATEGORY="system.essential"  
ISTATES="S 2"  
RSTATES="S 2"
```

**SEE ALSO**

**limit(1)**, **pkgmk(1)**, **pkgmap(4)**

*Application Packaging Developer's Guide*

**NOTES**

Developers may define their own installation parameters by adding a definition to this file. A developer-defined parameter must begin with a capital letter.

Trailing white space after any parameter value is ignored. For example, **VENDOR="AT&T"** is the same as **VENDOR="AT&T "**.

<b>NAME</b>	pkgmap – package contents description file
<b>DESCRIPTION</b>	<p><b>pkgmap</b> is an ASCII file that provides a complete listing of the package contents. It is automatically generated by <b>pkgmk(1)</b> using the information in the <b>prototype(4)</b> file.</p> <p>Each entry in <b>pkgmap</b> describes a single “deliverable object file.” A deliverable object file includes shell scripts, executable objects, data files, directories, and so forth. The entry consists of several fields of information, each field separated by a space. The fields are described below and must appear in the order shown.</p> <p><i>part</i>      An optional field designating the part number in which the object resides. A part is a collection of files and is the atomic unit by which a package is processed. A developer can choose the criteria for grouping files into a part (for example, based on class). If no value is defined in this field, part 1 is assumed.</p> <p><i>ftype</i>      A one-character field that indicates the file type. Valid values are:</p> <ul style="list-style-type: none"> <li><b>b</b>    block special device</li> <li><b>c</b>    character special device</li> <li><b>d</b>    directory</li> <li><b>e</b>    a file to be edited upon installation or removal (may be shared by several packages)</li> <li><b>f</b>    a standard executable or data file</li> <li><b>i</b>    installation script or information file</li> <li><b>l</b>    linked file</li> <li><b>p</b>    named pipe</li> <li><b>s</b>    symbolic link</li> <li><b>v</b>    volatile file (one whose contents are expected to change, like a log file)</li> <li><b>x</b>    an exclusive directory accessible only by this package</li> </ul> <p><i>class</i>      The installation class to which the file belongs. This name must contain only alphanumeric characters and be no longer than 12 characters. It is not specified if the <i>ftype</i> is <b>i</b> (information file).</p> <p><i>pathname</i>   <i>pathname</i> may contain variables of the form <i>\$variable</i> that support install-time configuration of the file. <i>variable</i> may be embedded in the pathname structure. (See <b>prototype(4)</b> for definitions of variable specifications.)</p> <p>Do not use the following reserved words in <i>pathname</i>, since they are applied by <b>pkgadd(1M)</b> using a different mechanism:</p> <ul style="list-style-type: none"> <li><b>PKG_INSTALL_ROOT</b></li> <li><b>BASEDIR</b></li> <li><b>CLIENT_BASEDIR</b></li> </ul> <p><i>major</i>      The major device number. The field is only specified for block or character special devices.</p>

<i>minor</i>	The minor device number. The field is only specified for block or character special devices.
<i>mode</i>	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 files, packaging information files, or non-installable files.  The mode can contain a variable specification. (See <b>prototype(4)</b> for definitions of variable specifications.)
<i>owner</i>	The owner of the file (for example, <b>bin</b> or <b>root</b> ). 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 files or non-installable files. It is used optionally with a package information file. If used, it indicates with what owner an installation script will be executed.  The owner can contain a variable specification. (See <b>prototype(4)</b> for definitions of variable specifications.)
<i>group</i>	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 files or non-installable files. It is used optionally with a package information file. If used, it indicates with what group an installation script will be executed.  The group can contain a variable specification. (See <b>prototype(4)</b> for definitions of variable specifications.)
<i>size</i>	The actual size of the file in bytes. This field is not specified for named pipes, special devices, directories or linked files.
<i>cksum</i>	The checksum of the file contents. This field is not specified for named pipes, special devices, directories, or linked files.
<i>modtime</i>	The time of last modification, as reported by the <b>stat(2)</b> function call. This field is not specified for named pipes, special devices, directories, or linked files.

Each **pkgmap** file must have one line that provides information about the number of parts and maximum size (in 512-byte blocks) of parts that make up the package. This line is in the following format:

```
:number_of_parts maximum_part_size
```

Lines that begin with “#” are comment lines and are ignored.

When files are saved during installation before they are overwritten, they are normally just copied to a temporary pathname. However, for files whose mode includes execute permission (but which are not editable), the existing version is linked to a temporary pathname and the original file is removed. This allows processes which are executing during installation to be overwritten.

**EXAMPLES**

The following is an example of a **pkgmap** file:

```

:2 500
1 i pkginfo 237 1179 541296672
1 b class1 /dev/diskette 17 134 0644 root other
1 c class1 /dev/rdiskette 17 134 0644 root other
1 d none bin 0755 root bin
1 f none bin/INSTALL 0755 root bin 11103 17954 541295535
1 f none bin/REMOVE 0755 root bin 3214 50237 541295541
1 l none bin/UNINSTALL=bin/REMOVE
1 f none bin/cmda 0755 root bin 3580 60325 541295567
1 f none bin/cmdb 0755 root bin 49107 51255 541438368
1 f class1 bin/cmdc 0755 root bin 45599 26048 541295599
1 f class1 bin/cmdd 0755 root bin 4648 8473 541461238
1 f none bin/cmde 0755 root bin 40501 1264 541295622
1 f class2 bin/cmdf 0755 root bin 2345 35889 541295574
1 f none bin/cmdg 0755 root bin 41185 47653 541461242
2 d class2 data 0755 root bin
2 p class1 data/apipe 0755 root other
2 d none log 0755 root bin
2 v none log/logfile 0755 root bin 41815 47563 541461333
2 d none save 0755 root bin
2 d none spool 0755 root bin
2 d none tmp 0755 root bin

```

**SEE ALSO**

**pkgmk(1)**, **pkgadd(1M)**, **stat(2)**, **pkginfo(4)**, **prototype(4)**

*Application Packaging Developer's Guide*

**NOTES**

The **pkgmap** file may contain only one entry per unique pathname.

<b>NAME</b>	platform – directory of files specifying supported platforms
<b>SYNOPSIS</b>	<b>.platform</b>
<b>DESCRIPTION</b>	<p>The Solaris 2.5 release includes the <b>.platform</b> directory, a new directory on the Solaris CD image. This directory contains files (created by SunSoft and Solaris OEMs) that define platform support. These files are generically referred to as <i>platform definition files</i>. They provide a means to map different platform types into a platform group.</p> <p>Platform definition files in the <b>.platform</b> directory are used by the installation software to ensure that software appropriate for the architecture of the system will be installed.</p> <p>SunSoft provides a platform definition file named <b>.platform/Solaris</b>. This file is the only one that can define platform groups to which other platform definition files can refer. For example, an OEM platform definition file can refer to any platform group specified in the Solaris platform definition file.</p> <p>Other platform definition files are delivered by OEMs. To avoid name conflicts, OEMs will name their platform definition file with an OEM-unique string. OEM's should use whatever string they use to make their package names unique. This unique string is often the OEM's stock symbol.</p> <p>Comments are allowed in a platform definition file. A "#" begins a comment and can be placed anywhere on a line.</p> <p>Platform definition files are composed of keyword-value pairs, and there are two kinds of stanzas in the file: platform group definitions and platform identifications.</p> <ul style="list-style-type: none"> <li>• Platform group definitions: <ul style="list-style-type: none"> <li>The keywords in a platform group definition stanza are:</li> <li><b>PLATFORM_GROUP</b> <p>The <b>PLATFORM_GROUP</b> keyword <i>must</i> be the first keyword in the platform group definition stanza. The value assigned to this keyword is the name of the platform group, for example:</p> <pre>PLATFORM_GROUP=sun4c</pre> <p>The <b>PLATFORM_GROUP</b> name is an arbitrary name assigned to a group of platforms. However, <b>PLATFORM_GROUP</b> typically equals the output of the <code>uname -m</code> command. <b>PLATFORM_GROUP</b> value cannot have white space and is limited to 256 ASCII characters.</p> </li> <li><b>INST_ARCH</b> <p>The instruction set architecture of all platforms in the platform group, for example:</p> <pre>INST_ARCH=sparc</pre> <p>The <b>INST_ARCH</b> keyword value must be the value returned by the <code>uname -p</code> command on all platforms in the platform group.</p> </li> </ul> </li> </ul>

- Platform identifications:

The keywords in a platform identification stanza are:

**PLATFORM\_NAME**

The **PLATFORM\_NAME** keyword *must* be the first keyword in the platform identification stanza. The **PLATFORM\_NAME** is the name assigned to the platform, for example:

```
PLATFORM_NAME=SUNW,SPARCstation-5
```

Typically, this name is the same as the value returned by the `uname -i` command on the machine, but it need not be the same.

The **PLATFORM\_NAME** value cannot have white space and is limited to 256 ASCII characters. If it contains parentheses, it must contain only balanced parentheses. For example, the string "foo(bar)foo" is a valid value for this keyword, but "foo(bar)" is not.

The other keywords in the platform identification stanza can be in any order, as long as the **PLATFORM\_NAME** keyword is first.

**PLATFORM\_ID**

The value returned by the `uname -i` command on the machine, for example:

```
PLATFORM_ID=SUNW,SPARCstation-5
```

**MACHINE\_TYPE**

The value returned by the `uname -m` command on the machine, for example:

```
MACHINE_TYPE=sun4c
```

**IN\_PLATFORM\_GROUP**

The platform group of which the platform is a member, for example:

```
IN_PLATFORM_GROUP=sun4c
```

The platform group name must be specified in the same file as the platform identification stanza or in the platform definition file with the name **.platform/Solaris**.

The **IN\_PLATFORM\_GROUP** keyword is optional. A platform doesn't have to belong to a platform group. If a platform isn't explicitly assigned to a platform group, it essentially forms its own platform group, where the platform group name is the **PLATFORM\_NAME** value.

The **IN\_PLATFORM\_GROUP** value typically equals the output of the `uname -m` command. **IN\_PLATFORM\_GROUP** value cannot have white space and is limited to 256 ASCII characters.

**INST\_ARCH**

The instruction set architecture of the platform, for example:

```
INST_ARCH=sparc
```

This field is only required if the platform does not belong to a platform group.

The **INST\_ARCH** keyword value must be the value returned by the `uname -p` command on all platforms in the platform group.

**COMPATIBILITY**

The installation program will remain compatible with the old Solaris CD format. If a Solaris CD image does not contain any platform definition files, the installation and upgrade programs will select the packages to be installed based on machine type (i.e., the value returned by the `uname -m` command).

**EXAMPLES**

1. The following example shows platform group definitions from the `.platform/Solaris` platform definition file.

```
#
PLATFORM_GROUP=sun4c
INST_ARCH=sparc
#
PLATFORM_GROUP=sun4d
INST_ARCH=sparc
#
PLATFORM_GROUP=sun4m
INST_ARCH=sparc
#
PLATFORM_GROUP=sun4u
INST_ARCH=sparc
```

2. The following example shows platform identification stanzas, which define systems that belong in a platform group, from the `.platform/Solaris` platform definition file.

```
#
PLATFORM_NAME=SUNW,Sun_4_20
PLATFORM_ID=SUNW,Sun_4_20
IN_PLATFORM_GROUP=sun4c
PLATFORM_NAME=SUNW,Sun_4_25
PLATFORM_ID=SUNW,Sun_4_25
IN_PLATFORM_GROUP=sun4c
#
PLATFORM_NAME=SUNW,SPARCstation-5
PLATFORM_ID=SUNW,SPARCstation-5
IN_PLATFORM_GROUP=sun4m
#
PLATFORM_NAME=SUNW,SPARCstation-10
PLATFORM_ID=SUNW,SPARCstation-10
IN_PLATFORM_GROUP=sun4m
```

**FILES**

The **.platform** directory must reside as `/cd_image/Solaris_vers/.platform`, where *cd\_image*

Is the path to the mounted Solaris CD (`/cdrom/cdrom0/s0` by default) or the path to a copy of the Solaris CD on a disk.

*Solaris\_vers*

Is the version of Solaris: e.g., Solaris\_2.5.

**NOTES**

Typically, a platform identification stanza contains either a **PLATFORM\_ID** or a **MACHINE\_TYPE** stanza, but *not* both. If both are specified, both must match for a platform to be identified as this platform type. Each platform identification stanza must contain either a **PLATFORM\_ID** value or a **MACHINE\_TYPE** value.

If a platform matches two different platform identification stanzas—one which matched on the value of **PLATFORM\_ID** and one which matched on the value of **MACHINE\_TYPE**, the one that matched on **PLATFORM\_ID** will take precedence.

The **.platform** directory is part of the Solaris CD image, whether that be the Solaris CD or a copy of the Solaris CD on a system's hard disk.



<b>NAME</b>	plot – graphics interface
<b>DESCRIPTION</b>	<p>Files of this format are interpreted for various devices by commands described in <b>plot(1B)</b>. A graphics file is a stream of plotting instructions. Each instruction consists of an ASCII letter usually followed by bytes of binary information. The instructions are executed in order. A point is designated by four bytes representing the <i>x</i> and <i>y</i> values; each value is a signed integer. The last designated point in an <b>l</b>, <b>m</b>, <b>n</b>, or <b>p</b> instruction becomes the “current point” for the next instruction.</p> <ul style="list-style-type: none"> <li><b>m</b> Move: the next four bytes give a new current point.</li> <li><b>n</b> Cont: draw a line from the current point to the point given by the next four bytes. See <b>plot(1B)</b>.</li> <li><b>p</b> Point: plot the point given by the next four bytes.</li> <li><b>l</b> Line: draw a line from the point given by the next four bytes to the point given by the following four bytes.</li> <li><b>t</b> Label: place the following ASCII string so that its first character falls on the current point. The string is terminated by a NEWLINE.</li> <li><b>a</b> Arc: the first four bytes give the center, the next four give the starting point, and the last four give the end point of a circular arc. The least significant coordinate of the end point is used only to determine the quadrant. The arc is drawn counter-clockwise.</li> <li><b>c</b> Circle: the first four bytes give the center of the circle, the next two the radius.</li> <li><b>e</b> Erase: start another frame of output.</li> <li><b>f</b> Linemod: take the following string, up to a NEWLINE, as the style for drawing further lines. The styles are “dotted,” “solid,” “longdashed,” “shortdashed,” and “dotdashed.” Effective only in <b>plot 4014</b> and <b>plot ver</b>.</li> <li><b>s</b> Space: the next four bytes give the lower left corner of the plotting area; the following four give the upper right corner. The plot will be magnified or reduced to fit the device as closely as possible.</li> </ul> <p>Space settings that exactly fill the plotting area with unity scaling appear below for devices supported by the filters of <b>plot(1B)</b>. The upper limit is just outside the plotting area.</p>

In every case the plotting area is taken to be square; points outside may be displayable on devices whose face is not square.

**4014**      **space(0, 0, 3120, 3120);**

**ver**        **space(0, 0, 2048, 2048);**

**300, 300s** **space(0, 0, 4096, 4096);**

**450**        **space(0, 0, 4096, 4096);**

**SEE ALSO**    **graph(1), plot(1B)**

<b>NAME</b>	power.conf – power management configuration information file
<b>SYNOPSIS</b>	<code>/etc/power.conf</code>
<b>DESCRIPTION</b>	<p>The <b>power.conf</b> file is used by the power management configuration program, <b>pmconfig</b>(1M), to initialize the settings for power management of the system.</p> <p>There are two types of entries in the <b>power.conf</b> file: <b>device management</b> entries and <b>system management</b> entries.</p>
<b>Device Management</b>	<p>Devices not appearing in this file will not be power managed without explicit configuration using the power management pseudo driver. See <b>pm</b>(7D). You should fully understand the power management framework before modifying device management entries in this file. Although inappropriate settings will not cause system damage, severe performance reduction may result. An entry in <b>power.conf</b> will be effective only if the driver for the device supports device power management.</p> <p>Device management entries consist of line by line listings of the devices to be configured. Each line is of the form:</p> <pre style="margin-left: 40px;"><i>device_name</i>           <i>threshold</i> . . .           <i>dependents</i> . . .</pre> <p>The fields must be in this order. Each line must contain a <i>device_name</i> field and a <i>threshold</i> field; it may also contain a <i>dependents</i> field. Fields and sub-fields are separated by white space (tabs or spaces). A line may be more than 80 characters. If a newline character is preceded by a backslash ('\') it will be treated as white space. Comment lines must begin with a hash character ('#').</p> <p>The <i>device_name</i> field specifies the device to be configured. <i>device_name</i> is either a pathname specifying the device special file or a relative pathname containing the name of the device special file. When using the latter format, instead of using the full pathname, it is possible to omit the portion of the pathname specifying the parent devices. This includes the leading '/'. Using this "relative" pathname format, the first device found with a full pathname containing <i>device_name</i> as its tail is matched. In either case, the leading <b>/devices</b> component of the pathname does not need to be specified.</p> <p>For example, a SCSI disk target with the following full path name:</p> <pre style="margin-left: 40px;"><b>/iommu@f,e000/sbus@f,e001/espdma@f,4000/esp@f,8000/sd@1,0</b></pre> <p>may also be specified as:</p> <pre style="margin-left: 40px;"><b>sbus@f,e000/espdma@f,4000/esp@f,8000/sd@1,0</b></pre> <p>or</p> <pre style="margin-left: 40px;"><b>esp@f,8000/sd@1,0</b></pre> <p>or</p> <pre style="margin-left: 40px;"><b>sd@1,0</b></pre> <p>The <i>threshold</i> field is used to configure the power manageable components of a device. These components represent entities within a device that may be power-managed separately. This field may contain as many integer values as the device has components. Each <i>threshold</i> time specifies the idle time in seconds before the respective component</p>

may be powered down. If there are fewer component *threshold* times than device components, the remaining components are not power managed. Use a value of `-1` to explicitly disable power-down for a component. At least one component *threshold* must be specified per device (in the file).

The *dependents* field may contain a list of *logical* dependents for this device. A *logical* dependent is a selected device that is not physically connected to the power managed device (for example, the display and the keyboard). A dependent device is one that must be idle and powered-down before the managed device can be powered down. The *dependents* field entries use the same format as the first field and are separated by white spaces. A device must previously have been configured before it can be used as a dependent.

Device power management entries for frame buffers are only effective when the X window system is not running. If either the Open Window or Common Desktop Environment window system is running, it takes over power management of the display devices that it is using.

### System Management

The system management entries control power management for the entire system. They are distinguished by the use of the special device names listed below.

Note that the following **autoshtutdown** entry is not intended to be hand edited, but to be maintained by **dtpower**.

If the *device\_name* field contains the special device name **autoshtutdown**, the *threshold* value specifies the *system idle time* (measured as discussed below) before the system may be shut down by **powerd(1M)**. The *threshold* value is followed by *start* and *finish* times (each in the format hh:mm) which specify the time period during which the system may be automatically shut down (see **powerd(1M)**). Following the *start* and *finish* times is the *behavior* field, consisting of one of the words **shutdown**, **noshutdown**, **autowakeup**, **default**, or **unconfigured**.

If the *behavior* field is **shutdown**, the system will be shut down automatically when it has been idle for the number of minutes specified in the *threshold* value and the time of day falls between the *start* and *finish* values.

If the *behavior* field is **noshutdown**, the system is never shut down automatically.

If the *behavior* field is **autowakeup** and the hardware has the capability to do autowakeup, the system is shut down as if the value were **shutdown** and the system will be restarted automatically the next time the time of day equals the *finish* time.

If the *behavior* field is **default**, the behavior of the system will depend upon its model. Desktop models that were first put into production after October 1, 1995 will behave as if the *behavior* field were set to **shutdown**. Desktop models first put into production before this date and server models will act as if the *behavior* field were set to **noshutdown**. The behavior is determined by a root node property named **energystar-v2**.

If the *behavior* field is **unconfigured**, the system will not be shut down automatically. If the system has just been installed or upgraded, the value of this field will be changed upon the next reboot. If the power management package has been added by hand, the **dtpower** utility must be run to set the correct autoshtutdown behavior.

If the *device\_name* field contains the special device name **statefile**, the *threshold* value specifies the location of the file used by **cpr**(7). The **cpr** module uses this file to record the state of the system prior to powering it down.

This entry has the following format:

**statefile**                      *pathname*

where *pathname* is the absolute pathname of a local ufs file. The pathname in the statefile entry must be the path to a local ufs file. It cannot be a symbolic link. If the file does not exist when it is time for a checkpoint to be taken, **cpr** will create it. All the directory components of the path must already exist.

The actual size required by **cpr** to checkpoint the system state at any given time depends on a variety of factors, including the size of the system's memory, the number of loadable drivers/modules in use, the number and type of processes running, and the amount of user memory that has been "locked down".

If **cpr** fails to complete a checkpoint due to insufficient space on the file system specified for the statefile, an explanatory message will be displayed on the console and written to the system log, and the system will be returned to its state prior to the checkpoint attempt.

It is recommended that the statefile be placed on a file system with at least 10 Mbytes of free space. In order that a newly installed system will have a statefile path which meets this requirement, a script run at boot time checks for the existence of the **power.conf** file. If the file exists but lacks a statefile entry, the script will create one using a simple method to determine the pathname. It first examines the free space in the root file system, and if there is sufficient space, an appropriate entry is added to **power.conf**. It then applies the same test to **/usr**, if it is a separate file system. If this also fails, it checks the file system of those remaining (if any) that has the largest number of free blocks. If all three of these checks fail, a message is displayed warning the user of the failure. If the **pathname** entry is created by the system, the final component of the name will be **.CPR**.

To further reduce the possibility of a checkpoint failure, the file system should have free space equivalent to at least one half of the system's memory (RAM). To modify the statefile location, edit the statefile entry in **power.conf**, replacing the existing path with the new one. After saving the file and exiting the editor, run the **pmconfig(1M)** command with no arguments.

Some types of application, such as proprietary data base packages, achieve higher performance by using Solaris system calls that lock a large number of user pages into memory. In such cases, the amount of space required for the **cpr** statefile should be increased by the total space of such locked down memory.

If the *device\_name* field contains the special device name **ttychars**, the *threshold* field will be interpreted as the maximum number of tty characters that can pass through the **ldterm** module with the system still considered to be idle. This value defaults to **0** if no entry is provided.

If the *device\_name* field contains the special device name **loadaverage**, the (floating point) *threshold* field will be interpreted as the maximum load average that can be seen with the system still considered to be idle. This value defaults to **0.04** if no entry is provided.

If the *device\_name* field contains the special device name **diskreads**, the *threshold* field will be interpreted as the maximum number of disk reads that can be performed by the system with the system still considered to be idle. This value defaults to **0** if no entry is provided.

If the *device\_name* field contains the special device name **nfsreqs**, the *threshold* field will be interpreted as the maximum number of NFS requests that can be sent or received by the system with the system still considered to be idle. Null requests, access requests, and get-tattr requests are excluded from this count. This value defaults to **0** if no entry is provided.

The values for tty characters, disk reads and NFS requests are determined by periodic sampling of the **kstat(3K)** interface. The thresholds for these events apply to a period extending into the past for *system idle time* minutes as specified in the **autoshutdown** entry described above.

The value for load average is also determined by periodic sampling of the **kstat** interface. The threshold for this value is an instantaneous one. The system will not be considered idle with respect to load average until *system idle time* minutes have passed with the sampled load average value not exceeding the threshold.

If the *device\_name* field contains the special device name **idlecheck**, the *device\_name* field must be followed by the pathname of a program to be executed to determine if the system is idle. If **autoshutdown** is enabled and the console keyboard, mouse, tty, CPU (as indicated by load average), network (as measured by NFS requests) and disk (as measured by read activity) have been idle for the amount of time specified in the *autoshutdown* entry specified above, and the time of day falls between the *start* and *finish* times, then this program will be executed to check for other idleness criteria. The value of the idle time specified in the above **autoshutdown** entry will be passed to the program in the environment variable **PM\_IDLETIME**. The process must terminate with an exit code that represents the number of minutes that the process considers the system to have been idle.

There is no default *idlecheck* entry. The default behavior is to consider only mouse, keyboard, tty, load average, NFS requests, and disk reads as indicators of non-idleness. To extend the definition of non-idleness, a shell script can be created that must exit with the number of minutes it considers the system to have been idle, according to its criteria. The path to this new script can then be stored in the *idlecheck* entry in **power.conf**.

#### EXAMPLES

power.conf file

The following is a sample **power.conf** file.

```
# This is a sample power management configuration file
# Fields must be separated by white space.
#
# Name           Threshold(s)  Logical Dependent(s)
/dev/kbd        1800
```

```

/dev/mouse      1800
/dev/fb         0 0          /dev/kbd /dev/mouse

#Example of a second display
/dev/fb1       0 0          /dev/kbd /dev/mouse

# This entry is maintained by dtpower
# This (default as of SunOS 5.5) entry causes the system to be
# shut down after 30 minutes of idle time if it is a model first
# shipped after Oct 1, 1995. Older models default to noshutdown.
#
#
#                               autosutdown in effect
# Auto-Shutdown      Idle(min)      Start/Finish(hh:mm)  Behavior
autosutdown         30      9:00 9:00                default

# Statefile              Path
statefile             /export/home/.CPR

# The idlecheck program is passed the autosutdown idle time entry in
# the environment variable $PM_IDLETIME and it must return the number of
# minutes the system has been idle (by its criteria) in its exit code.
idlecheck /home/critical/idlecheck

```

**idlecheck script**

The following is a sample idlecheck script.

```

#!/bin/sh
# This is a sample idlecheck script which considers the system
# not idle if user "critical" is logged in

critical='who | grep -w critical'
if [ "$critical" ]          # if "$critical" is not null string
then
    exit 0                  # not idle because critical logged in
else
    exit $PM_IDLETIME      # idle long enough
fi

```

**ATTRIBUTES**

See **attributes(5)** for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWpmr

**SEE ALSO**

**pmconfig(1M)**, **powerd(1M)**, **sys-unconfig(1M)**, **kstat(3K)**, **attributes(5)**, **cpr(7)**, **ldterm(7M)**, **pm(7D)**

*Writing Device Drivers***NOTES**

The default behavior for desktop models introduced after October 1, 1995 is to shut down after 30 minutes of idleness any time of day. The **dtpower** utility can be used to change the default.

The default behavior is mandated by the U.S. Government Environmental Protection Agency as a requirement for EnergyStar compliance. The user will be prompted to confirm this default at system installation reboot, or during the first boot after the system is unconfigured by **sys-unconfig**(1M).

The user may wish to use **dtpower** to set the **autoshtutdown start** time to the end of the normal work day, and to set the **autoshtutdown finish** time to the start of the normal work day.

The *physical* dependents are automatically included by the power manager and need not be specified.

The default **power.conf** file supports the standard hardware configuration. For each additional power manageable device (such as a second display), a new entry must be manually added to the **power.conf** file and **pmconfig**(1M) must be executed to activate the new change.

Frequently powering devices up and down may reduce device reliability, especially for devices not designed for power management. Do not place additional devices under power management unless the hardware documentation permits it. At this time most, SCSI hard disks are not power-manageable.



<b>NAME</b>	printers – user-configurable printer alias database
<b>SYNOPSIS</b>	<b>\$HOME/.printers</b>
<b>DESCRIPTION</b>	<p>The <b>\$HOME/.printers</b> file is a simplified version of the system <b>/etc/printers.conf</b> file (see <b>printers.conf(4)</b>). Users create the <b>\$HOME/.printers</b> file in their home directory. This optional file is customizable by the user.</p> <p>The <b>\$HOME/.printers</b> file performs the following functions:</p> <ol style="list-style-type: none"> <li>1. Sets personal aliases for all print commands.</li> <li>2. Sets the interest list for the <b>lpget</b>, <b>lpstat</b> and <b>cancel</b> commands. See <b>lpget(1M)</b>, <b>lpstat(1)</b> and <b>cancel(1)</b>.</li> <li>3. Sets the default printer for the <b>lp</b>, <b>lpr</b>, <b>lpq</b>, and <b>lprm</b> commands. See <b>lp(1)</b>, <b>lpr(1B)</b>, <b>lpq(1B)</b>, and <b>lprm(1B)</b>.</li> </ol>
<b>Entries</b>	<p>Use a line or full screen editor to create or modify the <b>\$HOME/.printers</b> file.</p> <p>Each entry in <b>\$HOME/.printers</b> describes one destination. Entries are one line consisting of two fields separated by either BLANKs or TABs and terminated by a NEWLINE. Format for an entry in <b>\$HOME/.printers</b> varies according to the purpose of the entry.</p> <p>Empty lines can be included for readability. Entries may continue on to multiple lines by adding a backslash ('\') as the last character in the line. The <b>\$HOME/.printers</b> file can include comments. Comments have a pound sign ('#') as the first character in the line, and are terminated by a NEWLINE.</p> <p><b>Setting Personal Aliases</b></p> <p>Specify the alias or aliases in the first field. Separate multiple aliases by a pipe sign (' '). Specify the destination in the second field. A destination names a printer or class of printers (see <b>lpadmin(1M)</b>). Specify the destination using atomic, POSIX-style (<i>server:destination</i>), or Federated Naming Service (FNS) (<i>.../service/printer/...</i>) names. See <b>printers.conf(4)</b> for information regarding the naming conventions for atomic and FNS names, and <b>standards(5)</b> for information regarding POSIX.</p> <p><b>Setting the Interest List for lpget, lpstat and cancel</b></p> <p>Specify <b>_all</b> in the first field. Specify the list of destinations for the interest list in the second field. Separate each destinations by a comma (','). Specify destinations using atomic, POSIX-style (<i>server:destination</i>), or FNS names (<i>.../service/printer/...</i>). See <b>printers.conf(4)</b> for information regarding the naming conventions for atomic and FNS names. This list of destinations may refer to an alias defined in <b>\$HOME/.printers</b>.</p> <p><b>Setting the Default Destination</b></p> <p>Specify <b>_default</b> in the first field. Specify the default destination in the second field. Specify the default destination using atomic, POSIX-style (<i>server:destination</i>), or FNS names (<i>.../service/printer/...</i>). See <b>printers.conf(4)</b> for information regarding the naming conventions for atomic and FNS names. The default destination may refer to an alias defined in <b>\$HOME/.printers</b>.</p>

**Locating Destination Information**

The print client commands locate destination information in a very specific order.

**Locating Destinations**

The print client commands locate destinations in the following order:

1. POSIX-style names.
2. Aliases in **\$HOME/.printers**.
3. Destinations in FNS.

**Locating the Interest List for lpstat, lpget and cancel**

The **lpget**, **lpstat** and **cancel** commands locate the interest list in the following order:

1. **\_all** list in **\$HOME/.printers**.
2. **\_all** list in **/etc/printers.conf**.
3. **\_all** list in FNS.

**Locating the Personal Default Destination**

The default destination is located differently depending on the command.

The **lp** command locates the default destination in the following order:

1. **lp** command's **-d destination** option.
2. **LPDEST** environment variable.
3. **PRINTER** environment variable.
4. **\_default** destination in **\$HOME/.printers**.
5. **\_default** destination in **/etc/printers.conf**.
6. **\_default** destination in FNS.

The **lpr**, **lpq**, and **lprm** commands locate the default destination in the following order:

1. **lpr** command's **-P destination** option.
2. **PRINTER** environment variable.
3. **LPDEST** environment variable.
4. **\_default** destination in **\$HOME/.printers**.
5. **\_default** destination in **/etc/printers.conf**.
6. **\_default** destination in FNS.

**EXAMPLES**

The following entry sets the interest list to destinations **ps**, **secure**, and **dog** at server **west** and **finance\_ps** at site **bldg2**.

```
_all          ps,secure,west:dog,site/bldg2/service/printer/finance_ps
```

The following entry sets the aliases **ps**, **lp**, and **lw** to **sparc\_printer**.

```
ps | lp | lw    sparc_printer
```

The following entry sets the alias **pcl** to **hplj** and sets it as the default destination.

```
pcl |_default hplj
```

The following entry sets the alias **secure** to destination **catalpa** at server **tabloid**.

```
secure tabloid:catalpa
```

The following entry sets the alias **insecure** to destination **legal\_ps** at site **bldg2**.

```
insecure site/bldg2/service/printer/legal_ps
```

<b>FILES</b>	<b>\$HOME/.printers</b>	User-configurable printer database.
	<b>/etc/printers.conf</b>	System printer configuration database.
	<b>printers.conf.byname</b>	NIS version of <b>/etc/printers.conf</b> .
	<b>fns.ctx_dir.domain</b>	NIS+ version of <b>/etc/printers.conf</b> .

**ATTRIBUTES** See **attributes(5)** for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWpcu

**SEE ALSO** **cancel(1)**, **lp(1)**, **lpq(1B)**, **lpr(1B)**, **lprm(1B)**, **lpstat(1)**, **lpadmin(1M)**, **lpget(1M)**, **printers.conf(4)**, **attributes(5)**, **fns(5)**, **standards(5)**

*System Administration Guide*

**NOTES** **\$HOME/.printers** is referenced by the printing commands before further name resolution is made in **/etc/printers.conf** or the name service. If the alias references a destination defined in **/etc/printers.conf**, it is possible that the destination is defined differently on different systems. This could cause output to be sent to an unintended destination if the user is logged in to a different system.

<b>NAME</b>	printers.conf – system printing configuration database
<b>SYNOPSIS</b>	<code>/etc/printers.conf</code>
<b>NIS</b>	<code>printers.conf.byname</code>
<b>NIS+</b>	<code>fns.ctx_dir.domain</code>
<b>DESCRIPTION</b>	<p>The <b>printers.conf</b> file is the system printing configuration database. System administrators use <b>printers.conf</b> to describe destinations for the print client commands and the print protocol adaptor. A destination names a printer or class of printers (see <b>lpadmin</b>(1M)). The LP print spooler uses private LP configuration data for represented in the <b>printers.conf</b> database.</p> <p><b>Entries</b></p> <p>Each entry in <b>printers.conf</b> describes one destination. Entries are one line consisting of any number of fields separated by colons (':') and terminated by a NEWLINE. The first field of each entry specifies the name of the destination and aliases to which the entry describes. Specify one or more names or aliases of the destination in this first field. Specify the destination using atomic names. POSIX-style names are not acceptable. See <b>standards</b>(5). Separate destination names by pipe signs (' ').</p> <p>Two destination names are reserved for special use in the first entry. Use <b>_all</b> to specify the interest list for <b>lpget</b>, <b>lpstat</b> and <b>cancel</b>. Use <b>_default</b> to specify the default destination.</p> <p>The remaining fields in an entry are <i>key=value</i> pairs. See <b>Specifying Configuration Options</b> for details regarding <i>key=value</i> pairs.</p> <p>Empty lines can be included for readability. Entries may continue on to multiple lines by adding a backslash ('\') as the last character in the line. <b>printers.conf</b> can include comments. Comments have a pound sign '#' as the first character in the line, and are terminated by a NEWLINE. Use the <b>lpset</b> command to create or modify <b>printers.conf</b> (see <b>lpset</b>(1M)). <i>Do not make changes in printers.conf using an editor.</i></p> <p><b>Specifying Configuration Options</b></p> <p><i>key=value</i> pairs are configuration options defined by the system administrator. <i>key</i> and <i>value</i> may be of arbitrary length. Separate <i>key</i> and <i>value</i> by the equal ('=') character.</p> <p><b>Client/Server Configuration Options</b></p> <p>The following client/server configuration options (represented as <i>key=value</i> pairs) are supported:</p> <p><b>bsdaddr=server,destination[,Solaris]</b></p> <p style="padding-left: 40px;">Sets the server and destination name. Sets if the client generates protocol extensions for use with the <b>lp</b> command (see <b>lp</b>(1)). <b>Solaris</b> specifies a Solaris print server extension. If <b>Solaris</b> is not specified, no protocol extensions are generated. <i>server</i> is the name of the host containing the queue for <i>destination</i>.</p>

<b>use=destination</b>	<i>destination</i> is the atomic name by which the server knows the destination. Sets the destination to continue searching for configuration information. <i>destination</i> is an atomic or Federated Naming Service (FNS) (.../ <b>service</b> / <b>printer</b> /...) name.
<b>all=destination_list</b>	Sets the interest list for the <b>lpget</b> , <b>lpstat</b> , and <b>cancel</b> commands. <i>destination_list</i> is a comma-separated (',') list of destinations. Specify <i>destination</i> using atomic or FNS names (.../ <b>service</b> / <b>printer</b> /...). See <b>lpget</b> (1M), <b>lpstat</b> (1), and <b>cancel</b> (1).

### General Server Options

The following general server configuration options (represented as *key=value* pairs) are supported:

<b>spooling-type=spooler[,version]</b>	Sets the type of spooler under which a destination is configured. Dynamically loads translation support for the back-end spooling system from <b>/usr/lib/print/bsd-adaptor/bsd_spooler.so[.version]</b> . Specify <i>spooler</i> as <b>lpsched</b> , <b>cascade</b> , or <b>test</b> . <b>lpsched</b> is used as a default for locally attached destinations. <b>cascade</b> is used as a default for destination spooled on a remote host. Use <b>test</b> for the test module to allow the capture of print requests. If using a versioned spooler module, <i>version</i> specifies the version of the translation module.
<b>spooling-type-path=dir_list</b>	Sets the location of translation support for the type of spooler defined by the <b>spooling-type</b> key. Locates translation support for the for the type of spooler under which a destination is configured. <i>dir_list</i> is a comma-separated (',') list of absolute pathnames to the directories used to locate translation support for the spooling system set by the <b>spooling-type</b> key.

### LP Server Options

The following LP configuration options (represented as *key=value* pairs) are supported:

<b>user-equivalence=true   false</b>	Sets whether or not usernames are considered equivalent when cancelling a print request submitted from a different host in a networked environment. <b>true</b> means that usernames are considered equivalent, and permits users to cancel a print requests submitted from a different host. <b>user-equivalence</b> is set to <b>false</b> by default. <b>false</b> means that usernames are not considered equivalent, and
--------------------------------------	---

does not permit users cancel a print request submitted from a different host. If **user-equivalence** is set to **false**, print requests can only be cancelled by the users on the host on which the print prequest was generated or by the super-user on the print server.

### Test Configuration Options

The following test configuration options (represented as *key=value* pairs) are supported:

<b>test-spooler-available=true   false</b>	Sets whether or not the protocol adaptor accepts connection requests to the test adaptor for the destination. <b>true</b> means that the protocol adaptor accepts connection requests to the test adaptor for the destination. <b>test-spooler-available</b> is set to <b>true</b> by default. <b>false</b> means that the protocol adaptor does not accept connection requests to the test adaptor for the destination.
<b>test-log=dir</b>	Sets the location of the log file generated by the test translation module. Specify <i>dir</i> as an absolute path-name.
<b>test-dir=dir</b>	Sets the directory to be used during execution of the test translation module. Specify <i>dir</i> as an absolute pathname.
<b>test-access=true   false</b>	Sets whether or not the requesting client has access to the test translation module. <b>true</b> means that the requesting client has access to the test translation module. <b>test-access</b> is set to <b>true</b> by default. <b>false</b> means that the the requesting client does not have access to the test translation module.
<b>test-accepting=true   false</b>	Sets whether or not the configured destination is accepting job submission requests. <b>true</b> means that the configured destination is accepting job submission requests. <b>test-accepting</b> is set to <b>true</b> by default. <b>false</b> means that the configured destination is not accepting job submission requests.
<b>test-restart=true   false</b>	Sets whether or not a protocol request to restart the destination will be honored or return an error. <b>true</b> means that a protocol request to restart the destination will be honored. <b>test-restart</b> is set to <b>true</b> by default. <b>false</b> means that a protocol request to restart the destination return an error.
<b>test-submit=true   false</b>	Sets whether or not a protocol request to submit a job to a destination will be honored or return an error. <b>true</b> means that a protocol request to submit a job to

	a destination will be honored. <b>test-submit</b> is set to <b>true</b> by default. <b>false</b> means that a protocol request to submit a job to a destination will not be honored.
<b>test-show-queue-file=</b> <i>file</i>	Sets the name of the file whose contents are to be returned as the result of a status query. Specify <i>file</i> as an absolute pathname.
<b>test-cancel-cancel-file=</b> <i>file</i>	Sets the name of the file whose contents are returned as the result of a cancellation request. Specify <i>file</i> as an absolute pathname.

### Locating Destination Information

The print client commands and the print protocol adaptor locate destination information in a very specific order.

#### Locating Destinations

The print client commands locate printers in the following order:

1. Aliases in `$HOME/.printers`.
3. Destinations in FNS.

#### Locating the Interest List for `lpstat`, `lpget` and `cancel`

The `lpget`, `lpstat` and `cancel` commands locate the interest list in the following order:

1. `_all` list in `$HOME/.printers`.
2. `_all` list in `/etc/printers.conf`.
3. `_all` list in FNS.

#### Locating the Personal Default Destination

The default destination is located differently depending on the command.

The `lp` command locates the default destination in the following order:

1. `lp` command's `-d destination` option.
2. `LPDEST` environment variable.
3. `PRINTER` environment variable.
4. `_default` destination in `$HOME/.printers`.
5. `_default` destination in `/etc/printers.conf`.
6. `_default` destination in FNS.

The `lpr`, `lpq`, and `lprm` commands locate the default destination in the following order:

1. `lpr` command's `-P destination` option.
2. `PRINTER` environment variable.
3. `LPDEST` environment variable.
4. `_default` destination in `$HOME/.printers`.
5. `_default` destination in `/etc/printers.conf`.
6. `_default` destination in FNS.

### Looking Up Destinations Using Atomic Names and FNS

Federated Naming Service (FNS) supports resolution of *composite* names spanning multiple naming systems. FNS supports several underlying naming services: NIS+, NIS, and files.

Atomic destination names are resolved using a specific search order. The order in which atomic destination names are resolved follows:

1. Atomic destination name in **/etc/printers.conf**.
2. Atomic destination name in Federated Naming Service (FNS) context.

The atomic destination name is searched for in the following FNS contexts in the order specified: **thisuser/service/printer**, **myorgunit/service/printer**, **thisorgunit/service/printer**.

In addition to these contexts, any subcontexts of these three contexts are also searched.

For example, if the target destination is **dept\_sparc**, and if **thisuser/service/printer** has a subcontext **color**, the following names will be looked up until one is found: **thisuser/service/printer/dept\_sparc**, **thisuser/service/printer/color/dept\_sparc**, **myorgunit/service/printer/dept\_sparc**, **thisorgunit/service/printer/dept\_sparc**.

3. If NIS is the underlying naming service and if the destination name is not found in **/etc/printers.conf** or the FNS contexts, the **printers.conf.byname** map is searched for the target destination.

FNS names such as **user/jsmith/service/printer/dept\_sparc** are looked up in FNS. There are no additional search rules or sources. The underlying naming service can be NIS+, NIS or files. See **fns(5)** for an overview of FNS. See **fns\_policies(5)** for an overview of FNS policies and defining names such as **thisuser** and **myorgunit**.

### EXAMPLES

The following entry sets the interest list for the **lpget**, **lpstat** and **cancel** commands to **printer1**, **printer2** and **printer3**.

```
_all:all=printer1,printer2,printer3
```

The following entry sets the server name to **server** and printer name to **ps\_printer** for destinations **printer1** and **ps**. It does not generate protocol extensions.

```
printer1 | ps:bsdaddr=server,ps_printer
```

The following entry sets the server name to **server** and destination name to **pcl\_printer**, for destination **printer2**. It also generates **Solaris** protocol extensions.

```
printer2:bsdaddr=server,pcl_printer,Solaris
```

The following entry sets the server name to **server** and destination name to **new\_printer**, for destination **printer3**. It also sets the **printer3** to continue searching for configuration information to printer **another\_printer**.

```
printer3:bsdaddr=server,new_printer:use=another_printer
```



The following entry sets the default destination to continue searching for configuration information to destination **printer1**.

**\_default:use=printer1**

**FILES**

<b>/etc/printers.conf</b>	System configuration database.
<b>\$HOME/.printers</b>	User-configurable printer database.
<b>printers.conf.byname</b> (NIS)	NIS version of <b>/etc/printers.conf</b> .
<b>fns.ctx_dir.domain</b>	NIS+ version of <b>/etc/printers.conf</b> .
<b>/usr/lib/print/bsd-adaptor/bsd_spooler.so*</b>	Spooler translation modules.
<b>/usr/lib/print/in.lpd</b>	BSD print protocol adapter.

**ATTRIBUTES** See **attributes(5)** for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWpcu

**SEE ALSO** **cancel(1)**, **lp(1)**, **lpq(1B)**, **lpr(1B)**, **lprm(1B)**, **lpstat(1)**, **in.lpd(1M)**, **lpadmin(1M)**, **lpget(1M)**, **lpset(1M)**, **printers(4)**, **attributes(5)**, **fns(5)**, **fns\_policies(5)**, **standards(5)**  
*System Administration Guide*

<b>NAME</b>	proc – /proc, the process file system
<b>DESCRIPTION</b>	<p><b>/proc</b> is a file system that provides access to the state of each process and light-weight process (lwp) in the system. The name of each entry in the <b>/proc</b> directory is a decimal number corresponding to a process-ID. These entries are themselves subdirectories. Access to process state is provided by additional files contained within each subdirectory; the hierarchy is described more completely below. In this document, “<b>/proc file</b>” refers to a non-directory file within the hierarchy rooted at <b>/proc</b>. The owner of each <b>/proc</b> file and subdirectory is determined by the user-ID of the process.</p> <p>Standard system calls are used to access <b>/proc</b> files: <b>open(2)</b>, <b>close(2)</b>, <b>read(2)</b>, and <b>write(2)</b> (including <b>readv(2)</b>, <b>writev(2)</b>, <b>pread(2)</b>, and <b>pwrite(2)</b>). Most files describe process state and can only be opened for reading. <b>ctl</b> and <b>lwpctl</b> (control) files permit manipulation of process state and can only be opened for writing. <b>as</b> (address space) files contain the image of the running process and can be opened for both reading and writing. An open for writing allows process control; a read-only open allows inspection but not control. In this document, we refer to the process as open for reading or writing if any of its associated <b>/proc</b> files is open for reading or writing.</p> <p>In general, more than one process can open the same <b>/proc</b> file at the same time. <i>Exclusive open</i> is an advisory mechanism provided to allow controlling processes to avoid collisions with each other. A process can obtain exclusive control of a target process, with respect to other cooperating processes, if it successfully opens any <b>/proc</b> file in the target process for writing (the <b>as</b> or <b>ctl</b> files, or the <b>lwpctl</b> file of any lwp) while specifying <b>O_EXCL</b> in the <b>open(2)</b>. Such an open will fail if the target process is already open for writing (that is, if an <b>as</b>, <b>ctl</b>, or <b>lwpctl</b> file is already open for writing). There can be any number of concurrent read-only opens; <b>O_EXCL</b> is ignored on opens for reading. It is recommended that the first open for writing by a controlling process use the <b>O_EXCL</b> flag; multiple controlling processes usually result in chaos.</p> <p>If a process opens one of its own <b>/proc</b> files for writing, the open succeeds regardless of <b>O_EXCL</b> and regardless of whether some other process has the process open for writing. Self-opens do not count when another process attempts an exclusive open. (A process cannot exclude a debugger by opening itself for writing and the application of a debugger cannot prevent a process from opening itself.) All self-opens for writing are forced to be close-on-exec (see the <b>F_SETFD</b> operation of <b>fcntl(2)</b>).</p> <p>Data may be transferred from or to any locations in the address space of the traced process by applying <b>lseek(2)</b> to position the <b>as</b> file at the virtual address of interest followed by <b>read(2)</b> or <b>write(2)</b> (or by using <b>pread(2)</b> or <b>pwrite(2)</b> for the combined operation). The address-map file <b>/proc/pid/map</b> can be read to determine the accessible areas (mappings) of the address space. I/O transfers may span contiguous mappings. An I/O request extending into an unmapped area is truncated at the boundary. A write request beginning at an unmapped virtual address fails with <b>EIO</b>; a read request beginning at an unmapped virtual address returns zero (an end-of-file indication).</p>

Information and control operations are provided through additional files. `<procfs.h>` contains definitions of data structures and message formats used with these files. Some of these definitions involve the use of sets of flags. The set types `sigset_t`, `fltset_t`, and `sysset_t` correspond, respectively, to signal, fault, and system call enumerations defined in `<sys/signal.h>`, `<sys/fault.h>`, and `<sys/syscall.h>`. Each set type is large enough to hold flags for its own enumeration. Although they are of different sizes, they have a common structure and can be manipulated by these macros:

```
prfillset(&set);           /* turn on all flags in set */
premptyset(&set);         /* turn off all flags in set */
praddset(&set, flag);     /* turn on the specified flag */
prdelset(&set, flag);     /* turn off the specified flag */
r = prismember(&set, flag); /* != 0 iff flag is turned on */
```

One of `prfillset()` or `premptyset()` must be used to initialize `set` before it is used in any other operation. `flag` must be a member of the enumeration corresponding to `set`.

Every process contains at least one *light-weight process*, or *lwp*. Each lwp represents a flow of execution that is independently scheduled by the operating system. All lwps in a process share its address space as well as many other attributes. Through the use of `lwpcntl` and `ctl` files as described below, it is possible to affect individual lwps in a process or to affect all of them at once, depending on the operation.

When the process has more than one lwp, a representative lwp is chosen by the system for certain process status files and control operations. The representative lwp is a stopped lwp only if all of the process's lwps are stopped; is stopped on an event of interest only if all of the lwps are so stopped (excluding `PR_SUSPENDED` lwps); is in a `PR_REQUESTED` stop only if there are no other events of interest to be found; or, failing everything else, is in a `PR_SUSPENDED` stop (implying that the process is deadlocked). See the description of the `status` file for definitions of stopped states. See the `PCSTOP` control operation for the definition of "event of interest".

The representative lwp remains fixed (it will be chosen again on the next operation) as long as all of the lwps are stopped on events of interest or are in a `PR_SUSPENDED` stop and the `PCRUN` control operation is not applied to any of them.

When applied to the process control file, every `/proc` control operation that must act on an lwp uses the same algorithm to choose which lwp to act upon. Together with synchronous stopping (see `PCSET`), this enables a debugger to control a multiple-lwp process using only the process-level status and control files if it so chooses. More fine-grained control can be achieved using the lwp-specific files.

## DIRECTORY STRUCTURE

At the top level, the directory `/proc` contains entries each of which names an existing process in the system. These entries are themselves directories. Except where otherwise noted, the files described below can be opened for reading only. In addition, if a process becomes a *zombie* (one that has exited but whose parent has not yet performed a `wait(2)` upon it), most of its associated `/proc` files disappear from the hierarchy; subsequent attempts to open them, or to read or write files opened before the process exited, will elicit the error `ENOENT`.

Although process state and consequently the contents of **/proc** files can change from instant to instant, a single **read(2)** of a **/proc** file is guaranteed to return a sane representation of state; that is, the read will be atomic with respect to the state of the process. No such guarantee applies to successive reads applied to a **/proc** file for a running process. In addition, atomicity is not guaranteed for I/O applied to the **as** (address-space) file for a running process or for a process whose address space contains memory shared by another running process.

A number of structure definitions are used to describe the files. These structures may grow by the addition of elements at the end in future releases of the system and it is not legitimate for a program to assume that they will not.

## STRUCTURE OF **/proc/pid**

A given directory **/proc/pid** contains the following entries. A process can use the invisible alias **/proc/self** if it wishes to open one of its own **/proc** files (invisible in the sense that the name “self” does not appear in a directory listing of **/proc** obtained from **ls(1)**, **getdents(2)**, or **readdir(3C)**).

- as** Contains the address-space image of the process; it can be opened for both reading and writing. **lseek(2)** is used to position the file at the virtual address of interest and then the address space can be examined or changed through **read(2)** or **write(2)** (or by using **pread(2)** or **pwrite(2)** for the combined operation).
- ctl** A write-only file to which structured messages are written directing the system to change some aspect of the process’s state or control its behavior in some way. The seek offset is not relevant when writing to this file. Individual lwps also have associated **lwpcctl** files in the lwp subdirectories. A control message may be written either to the process’s **ctl** file or to a specific **lwpcctl** file with operation-specific effects. The effect of a control message is immediately reflected in the state of the process visible through appropriate status and information files. The types of control messages are described in detail later. See **CONTROL MESSAGES**.
- status** Contains state information about the process and the representative lwp. The file contains a **pstatus** structure which contains an embedded **lwpstatus** structure for the representative lwp, as follows:

```
typedef struct pstatus {
    int          pr_flags;          /* flags (see below) */
    int          pr_nlwp;          /* number of lwps in the process */
    pid_t        pr_pid;           /* process id */
    pid_t        pr_ppid;          /* parent process id */
    pid_t        pr_pgid;          /* process group id */
    pid_t        pr_sid;           /* session id */
    id_t         pr_aslwpid;        /* lwp-id of the aslwp, if any */
    id_t         pr_agentid;        /* lwp-id of the agent lwp, if any */
    sigset_t     pr_sigpend;        /* set of process pending signals */
    uintptr_t    pr_brkbase;        /* virtual address of the process heap */
    size_t       pr_brksize;        /* size of the process heap, in bytes */
}
```

```

    uintptr_t    pr_stkbase;    /* virtual address of the process stack */
    size_t       pr_stksize;    /* size of the process stack, in bytes */
    timestruc_t  pr_utime;      /* process user cpu time */
    timestruc_t  pr_stime;      /* process system cpu time */
    timestruc_t  pr_cutime;     /* sum of children's user times */
    timestruc_t  pr_cstime;     /* sum of children's system times */
    sigset_t     pr_sigtrace;   /* set of traced signals */
    fltset_t     pr_fittrace;    /* set of traced faults */
    sysset_t     pr_sysentry;   /* set of system calls traced on entry */
    sysset_t     pr_sysexit;    /* set of system calls traced on exit */
    lwpstatus_t  pr_lwp;        /* status of the representative lwp */
} pstatus_t;

```

**pr\_flags** is a bit-mask holding the following process flags. For convenience, it also contains the lwp flags for the representative lwp, described later.

```

    PR_ISSYS      process is a system process (see PCSTOP).
    PR_VFORKPK    process is the parent of a vforked child (see PCWATCH).
    PR_FORK       process has its inherit-on-fork mode set (see PCSET).
    PR_RLC        process has its run-on-last-close mode set (see PCSET).
    PR_KLC        process has its kill-on-last-close mode set (see PCSET).
    PR_ASYNC      process has its asynchronous-stop mode set (see PCSET).
    PR_MSACCT     process has microstate accounting enabled (see PCSET).
    PR_MSFOURK    process microstate accounting is inherited on fork (see PCSET).
    PR_BPTADJ     process has its breakpoint adjustment mode set (see PCSET).
    PR_PTRACE     process has its ptrace-compatibility mode set (see PCSET).

```

**pr\_nlwps** is the total number of lwps in the process.

**pr\_pid**, **pr\_ppid**, **pr\_pgid**, and **pr\_sid** are, respectively, the process ID, the ID of the process's parent, the process's process group ID, and the process's session ID.

**pr\_aslwpid** is the lwp-ID for the "asynchronous signal lwp" (aslwp). It is zero if there is no aslwp in the process. The aslwp is the lwp designated to redirect asynchronous signals to other lwps in a multi-threaded process. See **signal(5)** for a description of the aslwp.

**pr\_agentid** is the lwp-ID for the **/proc** agent lwp (see the PCAGENT control operation). It is zero if there is no agent lwp in the process.

**pr\_sigpend** identifies asynchronous signals pending for the process.

**pr\_brkbase** is the virtual address of the process heap and **pr\_brksize** is its size in bytes. The address formed by the sum of these values is the process *break* (see **brk(2)**).

**pr\_stkbase** and **pr\_stksize** are, respectively, the virtual address of the process stack and its size in bytes. (Each lwp runs on a separate stack; the distinguishing characteristic of the process stack is that the operating system will grow it when necessary.)

**pr\_utime**, **pr\_stime**, **pr\_cutime**, and **pr\_cstime** are, respectively, the user CPU and system CPU time consumed by the process, and the cumulative user CPU and system CPU time consumed by the process's children, in seconds and nanoseconds.

**pr\_sigtrace** and **pr\_fitrace** contain, respectively, the set of signals and the set of hardware faults that are being traced (see PCSTRACE and PCSFAULT).

**pr\_sysentry** and **pr\_sysexit** contain, respectively, the sets of system calls being traced on entry and exit (see PCSEENTRY and PCSEXIT).

**pr\_lwp** contains the status information for the representative lwp:

```
typedef struct lwpstatus {
    int          pr_flags;          /* flags (see below) */
    id_t        pr_lwpid;          /* specific lwp identifier */
    short       pr_why;            /* reason for lwp stop, if stopped */
    short       pr_what;          /* more detailed reason */
    short       pr_cursig;         /* current signal, if any */
    siginfo_t   pr_info;          /* info associated with signal or fault */
    sigset_t    pr_lwppend;       /* set of signals pending to the lwp */
    sigset_t    pr_lwphold;       /* set of signals blocked by the lwp */
    struct sigaction pr_action;    /* signal action for current signal */
    stack_t     pr_altstack;      /* alternate signal stack info */
    uintptr_t   pr_oldcontext;    /* address of previous ucontext */
    short       pr_syscall;       /* system call number (if in syscall) */
    short       pr_nsysarg;       /* number of arguments to this syscall */
    int         pr_errno;         /* errno for failed syscall */
    long        pr_sysarg[PRSYSARGS]; /* arguments to this syscall */
    long        pr_rval1;         /* primary syscall return value */
    long        pr_rval2;         /* second syscall return value, if any */
    char        pr_clname[PRCLSZ]; /* scheduling class name */
    timestruc_t pr_tstamp;       /* real-time time stamp of stop */
    u_long      pr_instr;         /* current instruction */
    pgregset_t  pr_reg;           /* general registers */
    prfpregset_t pr_fpreg;       /* floating-point registers */
} lwpstatus_t;
```

**pr\_flags** is a bit-mask holding the following lwp flags. For convenience, it also contains the process flags, described previously.

<b>PR_STOPPED</b>	lwp is stopped.
<b>PR_ISTOP</b>	lwp is stopped on an event of interest (see PCSTOP).
<b>PR_DSTOP</b>	lwp has a stop directive in effect (see PCSTOP).
<b>PR_STEP</b>	lwp has a single-step directive in effect (see PCRUN).
<b>PR_ASLEEP</b>	lwp is in an interruptible sleep within a system call.
<b>PR_PCINVAL</b>	lwp's current instruction ( <b>pr_instr</b> ) is undefined.
<b>PR_ASLWP</b>	this is the asynchronous signal lwp for the process.
<b>PR_AGENT</b>	this is the <b>/proc</b> agent lwp for the process.

**pr\_lwpid** names the specific lwp.

**pr\_why** and **pr\_what** together describe, for a stopped lwp, the reason for the stop. Possible values of **pr\_why** and the associated **pr\_what** are:

**PR\_REQUESTED** indicates that the stop occurred in response to a stop directive, normally because PCSTOP was applied or because another lwp stopped on an event of interest and the asynchronous-stop flag (see PCSET) was not set for the process. **pr\_what** is unused in this case.

**PR\_SIGNALED** indicates that the lwp stopped on receipt of a signal (see PCSTRACE); **pr\_what** holds the signal number that caused the stop (for a newly-stopped lwp, the same value is in **pr\_cursig**).

**PR\_FAULTED** indicates that the lwp stopped on incurring a hardware fault (see PCSFAULT); **pr\_what** holds the fault number that caused the stop.

**PR\_SYSENTRY** and **PR\_SYSEXIT** indicate a stop on entry to or exit from a system call (see PCSENTRY and PCSEXIT); **pr\_what** holds the system call number.

**PR\_JOBCONTROL** indicates that the lwp stopped due to the default action of a job control stop signal (see **sigaction(2)**); **pr\_what** holds the stopping signal number.

**PR\_SUSPENDED** indicates that the lwp stopped due to internal synchronization of lwps within the process. **pr\_what** is unused in this case.

**pr\_cursig** names the current signal, that is, the next signal to be delivered to the lwp, if any. **pr\_info**, when the lwp is in a **PR\_SIGNALED** or **PR\_FAULTED** stop, contains additional information pertinent to the particular signal or fault (see `<sys/signinfo.h>`).

**pr\_lwppend** identifies any synchronous or directed signals pending for the lwp.

**pr\_lwphold** identifies those signals whose delivery is being blocked by the lwp (the signal mask).

**pr\_action** contains the signal action information pertaining to the current signal (see **sigaction(2)**); it is undefined if **pr\_cursig** is zero. **pr\_altstack** contains the alternate signal stack information for the lwp (see **sigaltstack(2)**).

**pr\_oldcontext**, if not zero, contains the address on the lwp stack of a **ucontext** structure describing the previous user-level context (see **ucontext(5)**). It is non-zero only if the lwp is executing in the context of a signal handler.

**pr\_syscall** is the number of the system call, if any, being executed by the lwp; it is non-zero if and only if the lwp is stopped on **PR\_SYSENTRY** or **PR\_SYSEXIT**, or is asleep within a system call (**PR\_ASLEEP** is set). If **pr\_syscall** is non-zero, **pr\_nsysarg** is the number of arguments to the system call and **pr\_sysarg** contains the actual arguments.

**pr\_rval1**, **pr\_rval2**, and **pr\_errno** are defined only if the lwp is stopped on **PR\_SYSEXIT** or if the **PR\_VFORKP** flag is set. If **pr\_errno** is zero, **pr\_rval1** and **pr\_rval2** contain the return values from the system call. Otherwise, **pr\_errno** contains the error number for the failing system call (see `<sys/errno.h>`).

**pr\_clname** contains the name of the lwp's scheduling class.

**pr\_tstamp**, if the lwp is stopped, contains a time stamp marking when the lwp stopped, in real time seconds and nanoseconds since an arbitrary time in the past.

**pr\_instr** contains the machine instruction to which the lwp's program counter refers. The amount of data retrieved from the process is machine-dependent. On SPARC based machines, it is a 32-bit word. On x86 based machines, it is a single byte. In general, the size is that of the machine's smallest instruction. If **PR\_PCINVAL** is set, **pr\_instr** is undefined; this occurs whenever the lwp is not stopped or when the program counter refers to an invalid virtual address.

**pr\_reg** is an array holding the contents of a stopped lwp's general registers.

On SPARC based machines the predefined constants **R\_G0 ... R\_G7**, **R\_O0 ... R\_O7**, **R\_L0 ... R\_L7**, **R\_I0 ... R\_I7**, **R\_PSR**, **R\_PC**, **R\_nPC**, **R\_Y**, **R\_WIM**, and **R\_TBR** can be used as indices to refer to the corresponding registers; previous register windows can be read from their overflow locations on the stack (however, see the **gwindows** file in the **/proc/pid/lwp/lwpid** subdirectory).

On x86 based machines, the predefined constants **SS**, **UESP**, **EFL**, **CS**, **EIP**, **ERR**, **TRAPNO**, **EAX**, **ECX**, **EDX**, **EBX**, **ESP**, **EBP**, **ESI**, **EDI**, **DS**, **ES**, **FS**, and **GS** can be used as indices to refer to the corresponding registers.

**pr\_fpreg** is a structure holding the contents of the floating-point registers.

If the lwp is not stopped, all register values are undefined.

## psinfo

Contains miscellaneous information about the process and the representative lwp needed by the **ps(1)** command. **psinfo** is accessible after a process becomes a *zombie*. The file contains a **psinfo** structure which contains an embedded **lwpsinfo** structure for the representative lwp, as follows:

```
typedef struct psinfo {
    int          pr_flag;           /* process flags */
    int          pr_nlwp;          /* number of lwps in the process */
    pid_t       pr_pid;           /* process id */
    pid_t       pr_ppid;          /* process id of parent */
    pid_t       pr_pgid;          /* process id of process group leader */
    pid_t       pr_sid;           /* session id */
    uid_t       pr_uid;           /* real user id */
    uid_t       pr_euid;          /* effective user id */
    gid_t       pr_gid;           /* real group id */
    gid_t       pr_egid;          /* effective group id */
    uintptr_t   pr_addr;          /* address of process */
    size_t      pr_size;           /* size of process image in Kbytes */
    size_t      pr_rssize;        /* resident set size in Kbytes */
    dev_t       pr_ttydev;        /* controlling tty device (or PRNODEV) */
    u_short     pr_pctcpu;        /* % of recent cpu time used by all lwps */
    u_short     pr_pctmem;        /* % of system memory used by process */
    timestruc_t pr_start;         /* process start time, from the epoch */
    timestruc_t pr_time;          /* cpu time for this process */
    timestruc_t pr_ctime;         /* cpu time for reaped children */
    char        pr_fname[PRFNSZ]; /* name of exec'ed file */
}
```



```

char    pr_psargs[PRARGSZ];    /* initial characters of arg list */
int     pr_wstat;              /* if zombie, the wait() status */
int     pr_argc;               /* initial argument count */
uintptr_t pr_argv;             /* address of initial argument vector */
uintptr_t pr_envp;             /* address of initial environment vector */
lwpsinfo_t pr_lwp;            /* information for representative lwp */
} psinfo_t;

```

Some of the entries in **psinfo**, such as **pr\_flag** and **pr\_addr**, refer to internal kernel data structures and should not be expected to retain their meanings across different versions of the operating system.

**pr\_pctcpu** and **pr\_pctmem** are 16-bit binary fractions in the range 0.0 to 1.0 with the binary point to the right of the high-order bit (1.0 == 0x8000). **pr\_pctcpu** is the summation over all lwps in the process.

**pr\_lwp** contains the **ps(1)** information for the representative lwp. If the process is a *zombie*, **pr\_nlwp** and **pr\_lwp.pr\_lwpid** are zero and the other fields of **pr\_lwp** are undefined:

```

typedef struct lwpsinfo {
int     pr_flag;                /* lwp flags */
id_t    pr_lwpid;              /* lwp id */
uintptr_t pr_addr;             /* internal address of lwp */
uintptr_t pr_wchan;            /* wait addr for sleeping lwp */
char    pr_stype;              /* synchronization event type */
char    pr_state;              /* numeric lwp state */
char    pr_sname;              /* printable character for pr_state */
char    pr_nice;               /* nice for cpu usage */
short   pr_syscall;            /* system call number (if in syscall) */
char    pr_oldpri;             /* pre-SVR4, low value is high priority */
char    pr_cpu;                /* pre-SVR4, cpu usage for scheduling */
int     pr_pri;                /* priority, high value = high priority */
u_short pr_pctcpu;            /* % of recent cpu time used by this lwp */
timestruc_t pr_start;         /* lwp start time, from the epoch */
timestruc_t pr_time;          /* cpu time for this lwp */
char    pr_clname[PRCLSZ];     /* scheduling class name */
char    pr_name[PRFNSZ];       /* name of system lwp */
processorid_t pr_onpro;         /* processor which last ran this lwp */
processorid_t pr_bindpro;       /* processor to which lwp is bound */
psetid_t pr_bindpset;         /* processor set to which lwp is bound */
} lwpsinfo_t;

```

Some of the entries in **lwpsinfo**, such as **pr\_flag**, **pr\_addr**, **pr\_wchan**, **pr\_stype**, **pr\_state**, and **pr\_name**, refer to internal kernel data structures and should not be expected to retain their meanings across different versions of the operating system.

**pr\_pctcpu** is a 16-bit binary fraction, as described above. It represents the CPU time used by the specific lwp. On a multi-processor machine, the maximum value is 1/N, where N is the number of CPUs.

- cred** Contains a description of the credentials associated with the process:
- ```
typedef struct prcred {
    uid_t    pr_euid;    /* effective user id */
    uid_t    pr_ruid;    /* real user id */
    uid_t    pr_suid;    /* saved user id (from exec) */
    gid_t    pr_egid;    /* effective group id */
    gid_t    pr_rgid;    /* real group id */
    gid_t    pr_sgid;    /* saved group id (from exec) */
    int      pr_ngroups; /* number of supplementary groups */
    gid_t    pr_groups[1]; /* array of supplementary groups */
} prcred_t;
```
- The array of associated supplementary groups in **pr\_groups** is of variable length; the **cred** file contains all of the supplementary groups. **pr\_ngroups** indicates the number of supplementary groups. (See also the PCSCRED control operation.)
- sigact** Contains an array of **sigaction** structures describing the current dispositions of all signals associated with the traced process (see **sigaction(2)**). Signal numbers are displaced by 1 from array indices, so that the action for signal number *n* appears in position *n*-1 of the array.
- auxv** Contains the initial values of the process's aux vector in an array of **auxv\_t** structures (see <**sys/auxv.h**>). The values are those that were passed by the operating system as startup information to the dynamic linker.
- ldt** This file exists only on x86 based machines. It is non-empty only if the process has established a local descriptor table (LDT). If non-empty, the file contains the array of currently active LDT entries in an array of elements of type **struct ssd**, defined in <**sys/sysi86.h**>, one element for each active LDT entry.
- map** Contains information about the virtual address map of the process. The file contains an array of **prmap** structures, each of which describes a contiguous virtual address region in the address space of the traced process:
- ```
typedef struct prmap {
    uintptr_t pr_vaddr;    /* virtual address of mapping */
    size_t    pr_size;    /* size of mapping in bytes */
    char      pr_mapname[PRMAPSZ]; /* name in /proc/pid/object */
    offset_t  pr_offset;  /* offset into mapped object, if any */
    int       pr_mflags;  /* protection and attribute flags */
    int       pr_pagesize; /* pagesize for this mapping in bytes */
} prmap_t;
```
- pr\_vaddr** is the virtual address of the mapping within the traced process and **pr\_size** is its size in bytes. **pr\_mapname**, if it does not contain a null string, contains the name of a file in the **object** directory (see below) that can be opened read-only to obtain a file descriptor for the mapped file associated with the mapping. This enables a debugger to

find object file symbol tables without having to know the real path names of the executable file and shared libraries of the process. **pr\_offset** is the 64-bit offset within the mapped file (if any) to which the virtual address is mapped.

**pr\_mflags** is a bit-mask of protection and attribute flags:

<b>MA_READ</b>	mapping is readable by the traced process.
<b>MA_WRITE</b>	mapping is writable by the traced process.
<b>MA_EXEC</b>	mapping is executable by the traced process.
<b>MA_SHARED</b>	mapping changes are shared by the mapped object.

A contiguous area of the address space having the same underlying mapped object may appear as multiple mappings due to varying read, write, and execute attributes. The underlying mapped object does not change over the range of a single mapping. An I/O operation to a mapping marked **MA\_SHARED** fails if applied at a virtual address not corresponding to a valid page in the underlying mapped object. A write to a **MA\_SHARED** mapping that is not marked **MA\_WRITE** fails. Reads and writes to private mappings always succeed. Reads and writes to unmapped addresses fail.

**pr\_pagesize** is the page size for the mapping, currently always the system pagesize.

- rmap** Contains information about the reserved address ranges of the process. The file contains an array of **prmap** structures, as defined above for the **map** file. Each structure describes a contiguous virtual address region in the address space of the traced process that is reserved by the system in the sense that an **mmap(2)** system call that does not specify **MAP\_FIXED** will not use any part of it for the new mapping. Examples of such reservations include the address ranges reserved for the process stack and the individual thread stacks of a multi-threaded process.
- cwd** A symbolic link to the process's current working directory (see **chdir(2)**). A **readlink(2)** of **/proc/pid/cwd** yields a null string. However, it can be opened, listed, and searched as a directory and can be the target of **chdir(2)**.
- root** A symbolic link to the process's root directory. **/proc/pid/root** can differ from the system root directory if the process or one of its ancestors executed **chroot(2)** as super-user. It has the same semantics as **/proc/pid/cwd**.
- fd** A directory containing references to the open files of the process. Each entry is a decimal number corresponding to an open file descriptor in the process.
- If an entry refers to a regular file, it can be opened with normal file system semantics but, to ensure that the controlling process cannot gain greater access than the controlled process, with no file access modes other than its read/write open modes in the controlled process. If an entry refers to a directory, it appears as a symbolic link and can be accessed with the same semantics as **/proc/pid/cwd**. An attempt to open any other type of entry fails with **EACCES**.

**object**

A directory containing read-only files with names corresponding to the **pr\_mapname** entries in the **map** and **pagedata** files. Opening such a file yields a file descriptor for the underlying mapped file associated with an address-space mapping in the process. The file name **a.out** appears in the directory as an alias for the process's executable file.

The **object** directory makes it possible for a controlling process to gain access to the object file and any shared libraries (and consequently the symbol tables) without having to know the actual path names of the executable files.

**pagedata**

Opening the page data file enables tracking of address space references and modifications on a per-page basis.

A **read(2)** of the page data file descriptor returns structured page data and atomically clears the page data maintained for the file by the system. That is to say, each read returns data collected since the last read; the first read returns data collected since the file was opened. When the call completes, the read buffer contains the following structure as its header and thereafter contains a number of section header structures and associated byte arrays that must be accessed by walking linearly through the buffer.

```
typedef struct prpageheader {
    timestruc_t  pr_tstamp;    /* real time stamp, time of read() */
    long         pr_nmap;     /* number of address space mappings */
    long         pr_npage;    /* total number of pages */
} prpageheader_t;
```

The header is followed by **pr\_nmap prasmap** structures and associated data arrays. The **prasmap** structure contains at least the following elements:

```
typedef struct prasmap {
    uintptr_t    pr_vaddr;    /* virtual address of mapping */
    size_t      pr_npage;    /* number of pages in mapping */
    char        pr_mapname[PRMAPSZ]; /* name in /proc/pid/object */
    offset_t    pr_offset;   /* offset into mapped object, if any */
    int         pr_mflags;   /* protection and attribute flags */
    int         pr_pagesize; /* pagesize for this mapping in bytes */
} prasmap_t;
```

Each section header is followed by **pr\_npage** bytes, one byte for each page in the mapping, plus 0-7 null bytes at the end so that the next **prasmap** structure begins on an eight-byte aligned boundary. Each data byte may contain these flags:

```
PG_REFERENCED    page has been referenced.
PG_MODIFIED      page has been modified.
```

If the read buffer is not large enough to contain all of the page data, the read fails with **E2BIG** and the page data is not cleared. The required size of the read buffer can be determined through **fstat(2)**. Application of **lseek(2)** to the page data file descriptor is ineffective; every read starts from the beginning of the file. Closing the page data file descriptor terminates the system overhead associated with collecting the data.

More than one page data file descriptor for the same process can be opened, up to a system-imposed limit per traced process. A read of one does not affect the data being collected by the system for the others. An open of the page data file will fail with ENOMEM if the system-imposed limit would be exceeded.

**watch** Contains an array of **prwatch** structures, one for each watched area established by the PCWATCH control operation. See PCWATCH for details.

**usage** Contains process usage information described by a **prusage** structure which contains at least the following fields:

```
typedef struct prusage {
    id_t      pr_lwpid;      /* lwp id. 0: process or defunct */
    int       pr_count;     /* number of contributing lwps */
    timestruc_t pr_tstamp;  /* real time stamp, time of read() */
    timestruc_t pr_create;  /* process/lwp creation time stamp */
    timestruc_t pr_term;    /* process/lwp termination time stamp */
    timestruc_t pr_rtime;   /* total lwp real (elapsed) time */
    timestruc_t pr_utime;   /* user level CPU time */
    timestruc_t pr_stime;   /* system call CPU time */
    timestruc_t pr_ttime;   /* other system trap CPU time */
    timestruc_t pr_tftime;  /* text page fault sleep time */
    timestruc_t pr_dftime;  /* data page fault sleep time */
    timestruc_t pr_kftime;  /* kernel page fault sleep time */
    timestruc_t pr_ltime;   /* user lock wait sleep time */
    timestruc_t pr_slptime; /* all other sleep time */
    timestruc_t pr_wtime;   /* wait-cpu (latency) time */
    timestruc_t pr_stoptime; /* stopped time */
    u_long    pr_minf;      /* minor page faults */
    u_long    pr_majf;      /* major page faults */
    u_long    pr_nswap;     /* swaps */
    u_long    pr_inblk;     /* input blocks */
    u_long    pr_oublk;     /* output blocks */
    u_long    pr_msnd;      /* messages sent */
    u_long    pr_mrcv;      /* messages received */
    u_long    pr_sigs;      /* signals received */
    u_long    pr_vctx;      /* voluntary context switches */
    u_long    pr_ictx;      /* involuntary context switches */
    u_long    pr_sysc;      /* system calls */
    u_long    pr_ioch;      /* chars read and written */
} prusage_t;
```

If microstate accounting has not been enabled for the process (see the PR\_MSACCT flag for the PCSET operation, below), the **usage** file contains only an estimate of times spent in the various states. The **usage** file is accessible after a process becomes a *zombie*.

<b>lstatus</b>	<p>Contains a <b>prheader</b> structure followed by an array of <b>lwpstatus</b> structures, one for each lwp in the process (see also <code>/proc/pid/lwp/lwpid/lwpstatus</code>, below). The <b>prheader</b> structure describes the number and size of the array entries that follow.</p> <pre>typedef struct prheader {     int          pr_nent;      /* number of entries */     int          pr_entsize;   /* size of each entry, in bytes */ } prheader_t;</pre> <p>The <b>lwpstatus</b> structure may grow by the addition of elements at the end in future releases of the system. Programs must use <b>pr_entsize</b> in the file header to index through the array. These comments apply to all <b>/proc</b> files that include a <b>prheader</b> structure (<b>lpsinfo</b> and <b>lusage</b>, below).</p>
<b>lpsinfo</b>	<p>Contains a <b>prheader</b> structure followed by an array of <b>lwpsinfo</b> structures, one for each lwp in the process. (See also <code>/proc/pid/lwp/lwpid/lwpsinfo</code>, below.)</p>
<b>lusage</b>	<p>Contains a <b>prheader</b> structure followed by an array of <b>prusage</b> structures, one for each lwp in the process plus an additional element at the beginning that contains the summation over all defunct lwps (lwps that once existed but no longer exist in the process). Excluding the <b>pr_lwpid</b>, <b>pr_tstamp</b>, <b>pr_create</b>, and <b>pr_term</b> entries, the entry-by-entry summation over all these structures is the definition of the process usage information obtained from the <b>usage</b> file. (See also <code>/proc/pid/lwp/lwpid/lwpusage</code>, below.)</p>
<b>lwp</b>	<p>A directory containing entries each of which names an lwp within the process. These entries are themselves directories containing additional files as described below.</p>
<b>STRUCTURE OF /proc/pid/lwp/lwpid</b>	<p>A given directory <code>/proc/pid/lwp/lwpid</code> contains the following entries:</p>
<b>lwpctl</b>	<p>Write-only control file. The messages written to this file affect the specific lwp rather than the representative lwp, as is the case for the process's <b>ctl</b> file.</p>
<b>lwpstatus</b>	<p>lwp-specific state information. This file contains the <b>lwpstatus</b> structure for the specific lwp as described above for the representative lwp in the process's <b>status</b> file.</p>
<b>lwpsinfo</b>	<p>lwp-specific <b>ps(1)</b> information. This file contains the <b>lwpsinfo</b> structure for the specific lwp as described above for the representative lwp in the process's <b>psinfo</b> file.</p>
<b>lwpusage</b>	<p>This file contains the <b>prusage</b> structure for the specific lwp as described above for the process's <b>usage</b> file.</p>
<b>gwindows</b>	<p>This file exists only on SPARC based machines. If it is non-empty, it contains a <b>gwindows_t</b> structure, defined in <code>&lt;sys/regset.h&gt;</code>, with the values of those SPARC register windows that could not be stored on the stack when the lwp stopped. Conditions under which register windows are not stored on the stack are: the stack pointer refers to nonexistent process memory or the stack pointer is improperly aligned. If the lwp is not stopped or if there are no register windows that could not be stored on the stack, the file</p>

is empty (the usual case).	
<b>xregs</b>	Extra state registers. The extra state register set is architecture dependent; this file is empty if the system does not support extra state registers. If the file is non-empty, it contains an architecture dependent structure of type <b>prxregset_t</b> , defined in <code>&lt;procfs.h&gt;</code> , with the values of the lwp's extra state registers. If the lwp is not stopped, all register values are undefined. See also the PCSXREG control operation, below.
<b>CONTROL MESSAGES</b>	<p>Process state changes are effected through messages written to a process's <b>ctl</b> file or to an individual lwp's <b>lwpctl</b> file. All control messages consist of a <b>long</b> that names the specific operation followed by additional data containing the operand, if any.</p> <p>Multiple control messages may be combined in a single <b>write(2)</b> (or <b>writew(2)</b>) to a control file, but no partial writes are permitted. That is, each control message, operation code plus operand, if any, must be presented in its entirety to the <b>write(2)</b> and not in pieces over several system calls. If a control operation fails, no subsequent operations contained in the same <b>write(2)</b> are attempted.</p> <p>Descriptions of the allowable control messages follow. In all cases, writing a message to a control file for a process or lwp that has terminated elicits the error ENOENT.</p>
<b>PCSTOP</b>	<p>When applied to the process control file, <b>PCSTOP</b> directs all lwps to stop and waits for them to stop, <b>PCDSTOP</b> directs all lwps to stop without waiting for them to stop, and <b>PCWSTOP</b> simply waits for all lwps to stop. When applied to an lwp control file, <b>PCSTOP</b> directs the specific lwp to stop and waits until it has stopped, <b>PCDSTOP</b> directs the specific lwp to stop without waiting for it to stop, and <b>PCWSTOP</b> simply waits for the specific lwp to stop. When applied to an lwp control file, <b>PCSTOP</b> and <b>PCWSTOP</b> complete when the lwp stops on an event of interest, immediately if already so stopped; when applied to the process control file, they complete when every lwp has stopped either on an event of interest or on a <b>PR_SUSPENDED</b> stop.</p> <p><b>PCTWSTOP</b> is identical to <b>PCWSTOP</b> except that it enables the operation to time out, to avoid waiting forever for a process or lwp that may never stop on an event of interest. <b>PCTWSTOP</b> takes a <b>long</b> operand specifying a number of milliseconds; the wait will terminate successfully after the specified number of milliseconds even if the process or lwp has not stopped; a timeout value of zero makes the operation identical to <b>PCWSTOP</b>.</p> <p>An "event of interest" is either a <b>PR_REQUESTED</b> stop or a stop that has been specified in the process's tracing flags (set by <b>PCSTRACE</b>, <b>PCSFAULT</b>, <b>PCSENTRY</b>, and <b>PCSEXIT</b>). <b>PR_JOBCONTROL</b> and <b>PR_SUSPENDED</b> stops are specifically not events of interest. (An lwp may stop twice due to a stop signal, first showing <b>PR_SIGNALLED</b> if the signal is traced and again showing <b>PR_JOBCONTROL</b> if the lwp is set running without clearing the signal.) If <b>PCSTOP</b> or <b>PCDSTOP</b> is applied to an lwp that is stopped, but not on an event of interest, the stop directive takes effect when the lwp is restarted by the competing mechanism. At that time, the lwp enters a <b>PR_REQUESTED</b> stop before executing any user-level code.</p>
<b>PCDSTOP</b>	
<b>PCWSTOP</b>	
<b>PCTWSTOP</b>	

A write of a control message that blocks is interruptible by a signal so that, for example, an **alarm(2)** can be set to avoid waiting forever for a process or lwp that may never stop on an event of interest. If **PCSTOP** is interrupted, the lwp stop directives remain in effect even though the **write(2)** returns an error. (Use of **PCTWSTOP** with a non-zero timeout is recommended over **PCWSTOP** with an **alarm(2)**.)

A system process (indicated by the **PR\_ISSYS** flag) never executes at user level, has no user-level address space visible through **/proc**, and cannot be stopped. Applying one of these operations to a system process or any of its lwps elicits the error **EBUSY**.

**PCRUN**

Make an lwp runnable again after a stop. This operation takes a **long** operand containing zero or more of the following flags:

**PRCSIG** clears the current signal, if any (see **PCCSIG**).

**PRCFAULT** clears the current fault, if any (see **PCCFAULT**).

**PRSTEP** directs the lwp to execute a single machine instruction. On completion of the instruction, a trace trap occurs. If **FLTTRACE** is being traced, the lwp stops; otherwise, it is sent **SIGTRAP**. If **SIGTRAP** is being traced and is not blocked, the lwp stops. When the lwp stops on an event of interest, the single-step directive is cancelled, even if the stop occurs before the instruction is executed. This operation requires hardware and operating system support and may not be implemented on all processors. It is implemented on SPARC and x86 based machines.

**PRSABORT** is meaningful only if the lwp is in a **PR\_SYSENTRY** stop or is marked **PR\_ASLEEP**; it instructs the lwp to abort execution of the system call (see **PCSENTRY** and **PCSEXIT**).

**PRSTOP** directs the lwp to stop again as soon as possible after resuming execution (see **PCDSTOP**). In particular, if the lwp is stopped on **PR\_SIGNALLED** or **PR\_FAULTED**, the next stop will show **PR\_REQUESTED**, no other stop will have intervened, and the lwp will not have executed any user-level code.

When applied to an lwp control file, **PCRUN** clears any outstanding directed-stop request and makes the specific lwp runnable. The operation fails with **EBUSY** if the specific lwp is not stopped on an event of interest or has not been directed to stop or if the agent lwp exists and this is not the agent lwp (see **PCAGENT**).

When applied to the process control file, a representative lwp is chosen for the operation as described for **/proc/pid/status**. The operation fails with **EBUSY** if the representative lwp is not stopped on an event of interest or has not been directed to stop or if the agent lwp exists. If **PRSTEP** or **PRSTOP** was requested, the representative lwp is made runnable and its outstanding directed-stop request is cleared; otherwise all outstanding directed-stop requests are cleared and, if it was stopped on an event of interest, the representative lwp is marked **PR\_REQUESTED**. If, as a consequence, all lwps are in the **PR\_REQUESTED** or **PR\_SUSPENDED** stop state, all lwps showing **PR\_REQUESTED** are made runnable.

**PCSTRACE**

Define a set of signals to be traced in the process. The receipt of one of these signals by an lwp causes the lwp to stop. The set of signals is defined using an operand **sigset\_t** contained in the control message. Receipt of **SIGKILL** cannot be traced; if specified, it is



	silently ignored.																						
	If a signal that is included in an lwp's held signal set (the signal mask) is sent to the lwp, the signal is not received and does not cause a stop until it is removed from the held signal set, either by the lwp itself or by setting the held signal set with PCSHOLD.																						
<b>PCCSIG</b>	The current signal, if any, is cleared from the specific or representative lwp.																						
<b>PCSSIG</b>	The current signal and its associated signal information for the specific or representative lwp are set according to the contents of the operand <b>siginfo</b> structure (see <code>&lt;sys/siginfo.h&gt;</code> ). If the specified signal number is zero, the current signal is cleared. The semantics of this operation are different from those of <b>kill(2)</b> in that the signal is delivered to the lwp immediately after execution is resumed (even if it is being blocked) and an additional PR_SIGNALED stop does not intervene even if the signal is traced. Setting the current signal to SIGKILL terminates the process immediately.																						
<b>PCKILL</b>	If applied to the process control file, a signal is sent to the process with semantics identical to those of <b>kill(2)</b> . If applied to an lwp control file, a directed signal is sent to the specific lwp. The signal is named in a <b>long</b> operand contained in the message. Sending SIGKILL terminates the process immediately.																						
<b>PCUNKILL</b>	A signal is deleted, that is, it is removed from the set of pending signals. If applied to the process control file, the signal is deleted from the process's pending signals. If applied to an lwp control file, the signal is deleted from the lwp's pending signals. The current signal (if any) is unaffected. The signal is named in a <b>long</b> operand in the control message. It is an error (EINVAL) to attempt to delete SIGKILL.																						
<b>PCSHOLD</b>	Set the set of held signals for the specific or representative lwp (signals whose delivery will be blocked if sent to the lwp). The set of signals is specified with a <b>sigset_t</b> operand. SIGKILL and SIGSTOP cannot be held; if specified, they are silently ignored.																						
<b>PCFAULT</b>	Define a set of hardware faults to be traced in the process. On incurring one of these faults, an lwp stops. The set is defined via the operand <b>fltset_t</b> structure. Fault names are defined in <code>&lt;sys/fault.h&gt;</code> and include the following. Some of these may not occur on all processors; there may be processor-specific faults in addition to these.																						
	<table border="0"> <tr> <td><b>FLTILL</b></td> <td>illegal instruction</td> </tr> <tr> <td><b>FLTPRIV</b></td> <td>privileged instruction</td> </tr> <tr> <td><b>FLTBP</b></td> <td>breakpoint trap</td> </tr> <tr> <td><b>FLTTRACE</b></td> <td>trace trap (single-step)</td> </tr> <tr> <td><b>FLTWATCH</b></td> <td>watchpoint trap</td> </tr> <tr> <td><b>FLTACCESS</b></td> <td>memory access fault (bus error)</td> </tr> <tr> <td><b>FLTBOUNDS</b></td> <td>memory bounds violation</td> </tr> <tr> <td><b>FLTIOVF</b></td> <td>integer overflow</td> </tr> <tr> <td><b>FLTIZDIV</b></td> <td>integer zero divide</td> </tr> <tr> <td><b>FLTTFPE</b></td> <td>floating-point exception</td> </tr> <tr> <td><b>FLTSTACK</b></td> <td>unrecoverable stack fault</td> </tr> </table>	<b>FLTILL</b>	illegal instruction	<b>FLTPRIV</b>	privileged instruction	<b>FLTBP</b>	breakpoint trap	<b>FLTTRACE</b>	trace trap (single-step)	<b>FLTWATCH</b>	watchpoint trap	<b>FLTACCESS</b>	memory access fault (bus error)	<b>FLTBOUNDS</b>	memory bounds violation	<b>FLTIOVF</b>	integer overflow	<b>FLTIZDIV</b>	integer zero divide	<b>FLTTFPE</b>	floating-point exception	<b>FLTSTACK</b>	unrecoverable stack fault
<b>FLTILL</b>	illegal instruction																						
<b>FLTPRIV</b>	privileged instruction																						
<b>FLTBP</b>	breakpoint trap																						
<b>FLTTRACE</b>	trace trap (single-step)																						
<b>FLTWATCH</b>	watchpoint trap																						
<b>FLTACCESS</b>	memory access fault (bus error)																						
<b>FLTBOUNDS</b>	memory bounds violation																						
<b>FLTIOVF</b>	integer overflow																						
<b>FLTIZDIV</b>	integer zero divide																						
<b>FLTTFPE</b>	floating-point exception																						
<b>FLTSTACK</b>	unrecoverable stack fault																						

	<p><b>FLTPAGE</b> recoverable page fault</p> <p>When not traced, a fault normally results in the posting of a signal to the lwp that incurred the fault. If an lwp stops on a fault, the signal is posted to the lwp when execution is resumed unless the fault is cleared by PCCFAULT or by the PRCFAULT option of PCRUN. <b>FLTPAGE</b> is an exception; no signal is posted. The <b>pr_info</b> field in the <b>lwpstatus</b> structure identifies the signal to be sent and contains machine-specific information about the fault.</p>								
<b>PCCFAULT</b>	<p>The current fault, if any, is cleared; the associated signal will not be sent to the specific or representative lwp.</p>								
<b>PCSENTRY PCSEXIT</b>	<p>These control operations instruct the process's lwps to stop on entry to or exit from specified system calls. The set of system calls to be traced is defined via an operand <b>sysset_t</b> structure.</p> <p>When entry to a system call is being traced, an lwp stops after having begun the call to the system but before the system call arguments have been fetched from the lwp. When exit from a system call is being traced, an lwp stops on completion of the system call just prior to checking for signals and returning to user level. At this point, all return values have been stored into the lwp's registers.</p> <p>If an lwp is stopped on entry to a system call (PR_SYSENTRY) or when sleeping in an interruptible system call (PR_ASLEEP is set), it may be instructed to go directly to system call exit by specifying the PRSABORT flag in a PCRUN control message. Unless exit from the system call is being traced, the lwp returns to user level showing EINTR.</p>								
<b>PCWATCH</b>	<p>Set or clear a watched area in the controlled process from a <b>prwatch</b> structure operand:</p> <pre>typedef struct prwatch {     uintptr_t    pr_vaddr;    /* virtual address of watched area */     size_t       pr_size;     /* size of watched area in bytes */     int          pr_wflags;   /* watch type flags */ } prwatch_t;</pre> <p><b>pr_vaddr</b> specifies the virtual address of an area of memory to be watched in the controlled process. <b>pr_size</b> specifies the size of the area, in bytes. <b>pr_wflags</b> specifies the type of memory access to be monitored as a bit-mask of the following flags:</p> <table border="0"> <tr> <td><b>WA_READ</b></td> <td>read access</td> </tr> <tr> <td><b>WA_WRITE</b></td> <td>write access</td> </tr> <tr> <td><b>WA_EXEC</b></td> <td>execution access</td> </tr> <tr> <td><b>WA_TRAPAFTER</b></td> <td>trap after the instruction completes</td> </tr> </table> <p>If <b>pr_wflags</b> is non-empty, a watched area is established for the virtual address range specified by <b>pr_vaddr</b> and <b>pr_size</b>. If <b>pr_wflags</b> is empty, any previously-established watched area starting at the specified virtual address is cleared; <b>pr_size</b> is ignored.</p> <p>A watchpoint is triggered when an lwp in the traced process makes a memory reference that covers at least one byte of a watched area and the memory reference is as specified in <b>pr_wflags</b>. When an lwp triggers a watchpoint, it incurs a watchpoint trap. If</p>	<b>WA_READ</b>	read access	<b>WA_WRITE</b>	write access	<b>WA_EXEC</b>	execution access	<b>WA_TRAPAFTER</b>	trap after the instruction completes
<b>WA_READ</b>	read access								
<b>WA_WRITE</b>	write access								
<b>WA_EXEC</b>	execution access								
<b>WA_TRAPAFTER</b>	trap after the instruction completes								

FLTWATCH is being traced, the lwp stops; otherwise, it is sent a SIGTRAP signal; if SIGTRAP is being traced and is not blocked, the lwp stops.

The watchpoint trap occurs before the instruction completes unless **WA\_TRAPAFTER** was specified, in which case it occurs after the instruction completes. If it occurs before completion, the memory is not modified. If it occurs after completion, the memory is modified (if the access is a write access).

**pr\_info** in the **lwpstatus** structure contains information pertinent to the watchpoint trap. In particular, the **si\_addr** field contains the virtual address of the memory reference that triggered the watchpoint, and the **si\_code** field contains one of **TRAP\_RWATCH**, **TRAP\_WWATCH**, or **TRAP\_XWATCH**, indicating read, write, or execute access, respectively. The **si\_trapafter** field is zero unless **WA\_TRAPAFTER** is in effect for this watched area; non-zero indicates that the current instruction is not the instruction that incurred the watchpoint trap. The **si\_pc** field contains the virtual address of the instruction that incurred the trap.

A watchpoint trap may be triggered while executing a system call that makes reference to the traced process's memory. The lwp that is executing the system call incurs the watchpoint trap while still in the system call. If it stops as a result, the **lwpstatus** structure contains the system call number and its arguments. If the lwp does not stop, or if it is set running again without clearing the signal or fault, the system call fails with **EFAULT**. If **WA\_TRAPAFTER** was specified, the memory reference will have completed and the memory will have been modified (if the access was a write access) when the watchpoint trap occurs.

If more than one of **WA\_READ**, **WA\_WRITE**, and **WA\_EXEC** is specified for a watched area, and a single instruction incurs more than one of the specified types, only one is reported when the watchpoint trap occurs. The precedence is **WA\_EXEC**, **WA\_READ**, **WA\_WRITE** (**WA\_EXEC** and **WA\_READ** take precedence over **WA\_WRITE**), unless **WA\_TRAPAFTER** was specified, in which case it is **WA\_WRITE**, **WA\_READ**, **WA\_EXEC** (**WA\_WRITE** takes precedence).

**PCWATCH** fails with **EINVAL** if an attempt is made to specify overlapping watched areas or if **pr\_wflags** contains flags other than those specified above. It fails with **ENOMEM** if an attempt is made to establish more watched areas than the system can support (the system can support thousands).

The child of a **vfork(2)** borrows the parent's address space. When a **vfork(2)** is executed by a traced process, all watched areas established for the parent are suspended until the child terminates or performs an **exec(2)**. Any watched areas established independently in the child are cancelled when the parent resumes after the child's termination or **exec(2)**. **PCWATCH** fails with **EBUSY** if applied to the parent of a **vfork(2)** before the child has terminated or performed an **exec(2)**. The **PR\_VFORKP** flag is set in the **pstatus** structure for such a parent process.

Certain accesses of the traced process's address space by the operating system are immune to watchpoints. The initial construction of a signal stack frame when a signal is delivered to an lwp will not trigger a watchpoint trap even if the new frame covers watched areas of the stack. Once the signal handler is entered, watchpoint traps occur

normally. On SPARC based machines, register window overflow and underflow will not trigger watchpoint traps, even if the register window save areas cover watched areas of the stack.

Watched areas are not inherited by child processes, even if the traced process's inherit-on-fork mode, `PR_FORK`, is set (see `PCSET`, below). All watched areas are cancelled when the traced process performs a successful `exec(2)`.

**PCSET**  
**PCUNSET**

**PCSET** sets one or more modes of operation for the traced process. **PCUNSET** unsets these modes. The modes to be set or unset are specified by flags in an operand **long** in the control message:

**PR\_FORK** (inherit-on-fork): When set, the process's tracing flags and its inherit-on-fork mode are inherited by the child of a `fork(2)`, `fork1(2)`, or `vfork(2)`. When unset, child processes start with all tracing flags cleared.

**PR\_RLC** (run-on-last-close): When set and the last writable `/proc` file descriptor referring to the traced process or any of its lwps is closed, all of the process's tracing flags and watched areas are cleared, any outstanding stop directives are canceled, and if any lwps are stopped on events of interest, they are set running as though `PCRUN` had been applied to them. When unset, the process's tracing flags and watched areas are retained and lwps are not set running on last close.

**PR\_KLC** (kill-on-last-close): When set and the last writable `/proc` file descriptor referring to the traced process or any of its lwps is closed, the process is terminated with `SIGKILL`.

**PR\_ASYNC** (asynchronous-stop): When set, a stop on an event of interest by one lwp does not directly affect any other lwp in the process. When unset and an lwp stops on an event of interest other than `PR_REQUESTED`, all other lwps in the process are directed to stop.

**PR\_MSACCT** (microstate accounting): When set, microstate accounting is enabled for the process. This allows the `usage` file to contain accurate values for the times the lwps spent in their various processing states. When unset (the default), the overhead of microstate accounting is avoided and the `usage` file can only contain an estimate of times spent in the various states.

**PR\_MSFOK** (inherit microstate accounting): When set, and microstate accounting is enabled for the process, microstate accounting will be enabled for future child processes. When unset, child processes start with microstate accounting disabled.

**PR\_BPTADJ** (breakpoint trap pc adjustment): On x86 based machines, a breakpoint trap leaves the program counter (the EIP) referring to the breakpointed instruction plus one byte. When `PR_BPTADJ` is set, the system will adjust the program counter back to the location of the breakpointed instruction when the lwp stops on a breakpoint. This flag has no effect on SPARC based machines, where breakpoint traps leave the program counter referring to the breakpointed instruction.

	<p><b>PR_PTRACE</b> (ptrace-compatibility): When set, a stop on an event of interest by the traced process is reported to the parent of the traced process via <b>wait(2)</b>, SIGTRAP is sent to the traced process when it executes a successful <b>exec(2)</b>, setuid/setgid flags are not honored for execs performed by the traced process, any exec of an object file that the traced process cannot read fails, and the process dies when its parent dies. This mode is deprecated; it is provided only to allow <b>ptrace(2)</b> to be implemented as a library function using <b>/proc</b>.</p> <p>It is an error (EINVAL) to specify flags other than those described above or to apply these operations to a system process. The current modes are reported in the <b>pr_flags</b> field of <b>/proc/pid/status</b> and <b>/proc/pid/lwp/lwpstatus</b>.</p>
<b>PCSREG</b>	<p>Set the general registers for the specific or representative lwp according to the operand <b>prgregset_t</b> structure.</p> <p>On SPARC based systems, only certain bits of the processor-status register (R_PS) can be modified by <b>PCSREG</b>: these include only the condition-code bits. Other privileged registers cannot be modified at all.</p> <p>On x86 based systems, only certain bits of the flags register (EFL) can be modified by <b>PCSREG</b>: these include the condition codes, direction-bit, and overflow-bit.</p> <p><b>PCSREG</b> fails with EBUSY if the lwp is not stopped on an event of interest.</p>
<b>PCSVADDR</b>	<p>Set the address at which execution will resume for the specific or representative lwp from the operand <b>long</b>. On SPARC based systems, both %pc and %npc are set, with %npc set to the instruction following the virtual address. On x86 based systems, only %eip is set. <b>PCSVADDR</b> fails with EBUSY if the lwp is not stopped on an event of interest.</p>
<b>PCSFREG</b>	<p>Set the floating-point registers for the specific or representative lwp according to the operand <b>prfpregset_t</b> structure. An error (EINVAL) is returned if the system does not support floating-point operations (no floating-point hardware and the system does not emulate floating-point machine instructions). <b>PCSFREG</b> fails with EBUSY if the lwp is not stopped on an event of interest.</p>
<b>PCSXREG</b>	<p>Set the extra state registers for the specific or representative lwp according to the architecture-dependent operand <b>prxregset_t</b> structure. An error (EINVAL) is returned if the system does not support extra state registers. <b>PCSXREG</b> fails with EBUSY if the lwp is not stopped on an event of interest.</p>
<b>PCAGENT</b>	<p>Create an agent lwp in the controlled process with register values from the operand <b>prgregset_t</b> structure (see PCSREG, above). The agent lwp is created in the stopped state showing PR_REQUESTED and with its held signal set (the signal mask) having all signals except SIGKILL and SIGSTOP blocked.</p> <p>The <b>PCAGENT</b> operation fails with EBUSY unless the process is fully stopped via <b>/proc</b>, that is, unless all of the lwps in the process are stopped either on events of interest or on PR_SUSPENDED, or are stopped on PR_JOBCONTROL and have been directed to stop via PCDSTOP. It fails with EBUSY if an agent lwp already exists. It fails with ENOMEM if</p>

system resources for creating new lwps have been exhausted.

Any PCRUN operation applied to the process control file or to the control file of an lwp other than the agent lwp fails with EBUSY as long as the agent lwp exists. The agent lwp must be caused to terminate by executing the `_lwp_exit(2)` system call before the process can be restarted.

Once the agent lwp is created, its lwp-ID can be found by reading the process status file. To facilitate opening the agent lwp's control and status files, the directory name `/proc/pid/lwp/agent` is accepted for lookup operations as an invisible alias for `/proc/pid/lwp/lwpid`, `lwpid` being the lwp-ID of the agent lwp (invisible in the sense that the name "agent" does not appear in a directory listing of `/proc/pid/lwp` obtained from `ls(1)`, `getdents(2)`, or `readdir(3C)`).

The purpose of the agent lwp is to perform operations in the controlled process on behalf of the controlling process: to gather information not directly available via `/proc` files, or in general to make the process change state in ways not directly available via `/proc` control operations. To make use of an agent lwp, the controlling process must be capable of making it execute system calls (specifically, the `_lwp_exit(2)` system call). The register values given to the agent lwp on creation are typically the registers of the representative lwp, so that the agent lwp can use its stack.

The agent lwp is not allowed to execute any variation of the `fork(2)`, `exec(2)`, or `_lwp_create(2)` system calls. Attempts to do so yield ENOTSUP to the agent lwp.

PCREAD  
PCWRITE

Read or write the target process's address space via a `piovec` structure operand:

```
typedef struct piovec {
    void      *pio_base;      /* buffer in controlling process */
    size_t    pio_len;       /* size of read/write request in bytes */
    off_t     pio_offset;    /* virtual address in target process */
} piovec_t;
```

These operations have the same effect as `pread(2)` and `pwrite(2)`, respectively, of the target process's address space file. The difference is that more than one PCREAD or PCWRITE control operation can be written to the control file at once, and they can be interspersed with other control operations in a single write to the control file. This is useful, for example, when planting many breakpoint instructions in the process's address space, or when stepping over a breakpointed instruction. Unlike `pread(2)` and `pwrite(2)`, no provision is made for partial reads or writes; if the operation cannot be performed completely, it fails with EIO.

PCNICE

The traced process's `nice(2)` value is incremented by the amount in the operand `long`. Only the super-user may better a process's priority in this way, but any user may lower the priority. This operation is not meaningful for all scheduling classes.

PCSCRED

Set the target process credentials to the values contained in the `prcred_t` structure operand (see `/proc/pid/cred`). The effective, real, and saved user-IDs and group-IDs of the target process are set. The target process's supplementary groups are not changed; the `pr_ngroups` and `pr_groups` members of the structure operand are ignored. Only the

PROGRAMMING  
NOTES

super-user may perform this operation; for all others it fails with EPERM.

For security reasons, except for the **psinfo**, **usage**, **lpsinfo**, **lusage**, **lwpsinfo**, and **lwpusage** files, which are world-readable, and except for the super-user, an open of a **/proc** file fails unless both the user-ID and group-ID of the caller match those of the traced process and the process's object file is readable by the caller. Except for the world-readable files just mentioned, files corresponding to **setuid** and **setgid** processes can be opened only by the super-user.

Even if held by the super-user, an open process or lwp file descriptor (other than file descriptors for the world-readable files) becomes invalid if the traced process performs an **exec(2)** of a **setuid/setgid** object file or an object file that the traced process cannot read. Any operation performed on an invalid file descriptor, except **close(2)**, fails with EAGAIN. In this situation, if any tracing flags are set and the process or any lwp file descriptor is open for writing, the process will have been directed to stop and its run-on-last-close flag will have been set (see PCSET). This enables a controlling process (if it has permission) to reopen the **/proc** files to get new valid file descriptors, close the invalid file descriptors, unset the run-on-last-close flag (if desired), and proceed. Just closing the invalid file descriptors causes the traced process to resume execution with all tracing flags cleared. Any process not currently open for writing via **/proc**, but that has left-over tracing flags from a previous open, and that executes a **setuid/setgid** or unreadable object file, will not be stopped but will have all its tracing flags cleared.

To wait for one or more of a set of processes or lwps to stop or terminate, **/proc** file descriptors (other than those obtained by opening the **cwd** or **root** directories or by opening files in the **fd** or **object** directories) can be used in a **poll(2)** system call. When requested and returned, either of the polling events POLLPRI or POLLWRNORM indicates that the process or lwp stopped on an event of interest. Although they cannot be requested, the polling events POLLHUP, POLLERR, and POLLNVAL may be returned. POLLHUP indicates that the process or lwp has terminated. POLLERR indicates that the file descriptor has become invalid. POLLNVAL is returned immediately if POLLPRI or POLLWRNORM is requested on a file descriptor referring to a system process (see PCSTOP). The requested events may be empty to wait simply for termination.

## FILES

<b>/proc</b>	directory (list of processes)
<b>/proc/pid</b>	specific process directory
<b>/proc/self</b>	alias for a process's own directory
<b>/proc/pid/as</b>	address space file
<b>/proc/pid/ctl</b>	process control file
<b>/proc/pid/status</b>	process status
<b>/proc/pid/lstatus</b>	array of lwp status structs
<b>/proc/pid/psinfo</b>	process <b>ps(1)</b> info
<b>/proc/pid/lpsinfo</b>	array of lwp <b>ps(1)</b> info structs
<b>/proc/pid/map</b>	address space map
<b>/proc/pid/rmap</b>	reserved address map
<b>/proc/pid/cred</b>	process credentials
<b>/proc/pid/sigact</b>	process signal actions

<code>/proc/pid/auxv</code>	process aux vector
<code>/proc/pid/ldt</code>	process LDT (x86 only)
<code>/proc/pid/usage</code>	process usage
<code>/proc/pid/lusage</code>	array of lwp usage structs
<code>/proc/pid/pagedata</code>	process page data
<code>/proc/pid/watch</code>	active watchpoints
<code>/proc/pid/cwd</code>	symlink to the current working directory
<code>/proc/pid/root</code>	symlink to the root directory
<code>/proc/pid/fd</code>	directory (list of open files)
<code>/proc/pid/fd/*</code>	aliases for process's open files
<code>/proc/pid/object</code>	directory (list of mapped files)
<code>/proc/pid/object/a.out</code>	alias for process's executable file
<code>/proc/pid/object/*</code>	aliases for other mapped files
<code>/proc/pid/lwp</code>	directory (list of lwps)
<code>/proc/pid/lwp/lwpid</code>	specific lwp directory
<code>/proc/pid/lwp/agent</code>	alias for the agent lwp directory
<code>/proc/pid/lwp/lwpid/lwpctl</code>	lwp control file
<code>/proc/pid/lwp/lwpid/lwpstatus</code>	lwp status
<code>/proc/pid/lwp/lwpid/lwpsinfo</code>	lwp <b>ps</b> (1) info
<code>/proc/pid/lwp/lwpid/lwpusage</code>	lwp usage
<code>/proc/pid/lwp/lwpid/gwindows</code>	register windows (SPARC only)
<code>/proc/pid/lwp/lwpid/xregs</code>	extra state registers

**SEE ALSO**

**ls**(1), **ps**(1), **\_lwp\_create**(2), **\_lwp\_exit**(2), **alarm**(2), **brk**(2), **chdir**(2), **chroot**(2), **close**(2), **creat**(2), **dup**(2), **exec**(2), **fcntl**(2), **fork**(2), **fork1**(2), **fstat**(2), **getdents**(2), **kill**(2), **lseek**(2), **mmap**(2), **nice**(2), **open**(2), **poll**(2), **pread**(2), **ptrace**(2), **pwrite**(2), **read**(2), **readlink**(2), **readv**(2), **sigaction**(2), **sigaltstack**(2), **vfork**(2), **wait**(2), **write**(2), **writev**(2), **readdir**(3C), **siginfo**(5), **signal**(5), **ucontext**(5)

**DIAGNOSTICS**

Errors that can occur in addition to the errors normally associated with file system access:

<b>ENOENT</b>	The traced process or lwp has terminated after being opened.
<b>EIO</b>	A <b>write</b> (2) was attempted at an illegal address in the traced process.
<b>EBUSY</b>	PCSTOP, PCDSTOP, PCWSTOP, or PCTWSTOP was applied to a system process; an exclusive <b>open</b> (2) was attempted on a <b>/proc</b> file for a process already open for writing; PCRUN, PCSREG, PCSVADDR, PCSFPREG, or PCSXREG was applied to a process or lwp not stopped on an event of interest; an attempt was made to mount <b>/proc</b> when it was already mounted; PCAGENT was applied to a process that was not fully stopped or that already had an agent lwp.
<b>EPERM</b>	Someone other than the super-user issued the PCSCRED operation; someone other than the super-user attempted to better a process's priority by applying PCNICE.
<b>ENOSYS</b>	An attempt was made to perform an unsupported operation (such as <b>creat</b> (2), <b>link</b> (2), or <b>unlink</b> (2)) on an entry in <b>/proc</b> .
<b>EINVAL</b>	In general, this means that some invalid argument was supplied to a system



call. A non-exhaustive list of conditions eliciting this error includes: a control message operation code is undefined; an out-of-range signal number was specified with PCSSIG, PCKILL, or PCUNKILL; SIGKILL was specified with PCUNKILL; PCSFPREG was applied on a system that does not support floating-point operations; PCSXREG was applied on a system that does not support extra state registers.

- ENOMEM** The system-imposed limit on the number of page data file descriptors was reached on an open of **/proc/pid/pagedata**; an attempt was made with PCWATCH to establish more watched areas than the system can support; the PCAGENT operation was issued when the system was out of resources for creating lwps.
- E2BIG** Data to be returned in a **read(2)** of the page data file exceeds the size of the read buffer provided by the caller.
- EINTR** A signal was received by the controlling process while waiting for the traced process or lwp to stop via PCSTOP, PCWSTOP, or PCTWSTOP.
- EAGAIN** The traced process has performed an **exec(2)** of a setuid/setgid object file or of an object file that it cannot read; all further operations on the process or lwp file descriptor (except **close(2)**) elicit this error.

**NOTES** Descriptions of structures in this document include only interesting structure elements, not filler and padding fields, and may show elements out of order for descriptive clarity. The actual structure definitions are contained in **<procfs.h>**.

**BUGS** Because the old **ioctl(2)**-BASED version of **/proc** is currently supported for binary compatibility with old applications, the top-level directory for a process, **/proc/pid**, is not world-readable, but it is world-searchable. Thus, anyone can open **/proc/pid/psinfo** even though **ls(1)** applied to **/proc/pid** will fail for anyone but the owner or the super-user. Support for the old **ioctl(2)**-BASED version of **/proc** will be dropped in a future release, at which time the top-level directory for a process will be made world-readable.

On SPARC based machines, the types **gregset\_t** and **fpregset\_t** defined in **<sys/regset.h>** are similar to but not the same as the types **prgregset\_t** and **prfpregset\_t** defined in **<procfs.h>**.

<b>NAME</b>	profile – setting up an environment for user at login time				
<b>SYNOPSIS</b>	<b>/etc/profile</b> <b>\$HOME/.profile</b>				
<b>DESCRIPTION</b>	<p>All users who have the shell, <b>sh(1)</b>, as their login command have the commands in these files executed as part of their login sequence.</p> <p><b>/etc/profile</b> allows the system administrator to perform services for the entire user community. Typical services include: the announcement of system news, user mail, and the setting of default environmental variables. It is not unusual for <b>/etc/profile</b> to execute special actions for the <b>root</b> login or the <b>su</b> command.</p> <p>The file <b>\$HOME/.profile</b> is used for setting per-user exported environment variables and terminal modes. The following example is typical (except for the comments):</p> <pre> # Make some environment variables global export MAIL PATH TERM # Set file creation mask umask 022 # Tell me when new mail comes in MAIL=/var/mail/\$LOGNAME # Add my /usr/usr/bin directory to the shell search sequence PATH=\$PATH:\$HOME/bin # Set terminal type TERM=\${L0:-u/n/k/n/o/w/n} # gnar.invalid while : do     if [ -f \${TERMINFO:-/usr/share/lib/terminfo}/?/\$TERM ]     then break     elif [ -f /usr/share/lib/terminfo/?/\$TERM ]     then break     else echo "invalid term \$TERM" 1&gt;&amp;2     fi     echo "terminal: \c"     read TERM done # Initialize the terminal and set tabs # Set the erase character to backspace stty erase ^H' echoe </pre>				
<b>FILES</b>	<table border="0"> <tr> <td style="padding-right: 20px;"><b>\$HOME/.profile</b></td> <td>user-specific environment</td> </tr> <tr> <td><b>/etc/profile</b></td> <td>system-wide environment</td> </tr> </table>	<b>\$HOME/.profile</b>	user-specific environment	<b>/etc/profile</b>	system-wide environment
<b>\$HOME/.profile</b>	user-specific environment				
<b>/etc/profile</b>	system-wide environment				

**SEE ALSO** **env(1), login(1), mail(1), sh(1), stty(1), tput(1), su(1M), terminfo(4), environ(5), term(5)**  
*Solaris Advanced User's Guide*

**NOTES** Care must be taken in providing system-wide services in **/etc/profile**. Personal **.profile** files are better for serving all but the most global needs.

<b>NAME</b>	protocols – protocol name database
<b>SYNOPSIS</b>	<b>/etc/inet/protocols</b> <b>/etc/protocols</b>
<b>DESCRIPTION</b>	<p>The <b>protocols</b> file is a local source of information regarding the known protocols used in the DARPA Internet. The protocols file can be used in conjunction with or instead of other protocols sources, including the NIS maps “protocols.byname” and “protocols.bynumber” and the NIS+ table “protocols”. Programs use the <b>getprotobyname(3N)</b> routine to access this information.</p> <p>The protocols file has one line for each protocol. The line has the following format:</p> <p style="text-align: center;"><i>official-protocol-name      protocol-number aliases</i></p> <p>Items are separated by any number of blanks and/or TAB characters. A ‘#’ indicates the beginning of a comment; characters up to the end of the line are not interpreted by routines which search the file. Protocol names may contain any printable character other than a field delimiter, NEWLINE, or comment character.</p>
<b>EXAMPLES</b>	<p>The following is a sample database:</p> <pre># # Internet (IP) protocols # ip      0      IP      # internet protocol, pseudo protocol number icmp    1      ICMP    # internet control message protocol ggp     3      GGP     # gateway-gateway protocol tcp     6      TCP     # transmission control protocol pup     12     PUP     # PARC universal packet protocol udp     17     UDP     # user datagram protocol</pre>
<b>FILES</b>	<b>/etc/nsswitch.conf</b> configuration file for name-service switch
<b>SEE ALSO</b>	<b>getprotobyname(3N)</b> , <b>nsswitch.conf(4)</b>
<b>NOTES</b>	<b>/etc/inet/protocols</b> is the official SVR4 name of the <b>protocols</b> file. The symbolic link <b>/etc/protocols</b> exists for BSD compatibility.

<b>NAME</b>	prototype – package information file
<b>DESCRIPTION</b>	<p><b>prototype</b> is an ASCII file used to specify package information. Each entry in the file describes a single deliverable object. An object may be a data file, directory, source file, executable object, and so forth. This file is generated by the package developer.</p> <p>Entries in a <b>prototype</b> file consist of several fields of information separated by white space. Comment lines begin with a “#” and are ignored. The fields are described below and must appear in the order shown.</p> <p><i>part</i> An optional field designating the part number in which the object resides. A part is a collection of files and is the atomic unit by which a package is processed. A developer can choose criteria for grouping files into a part (for example, based on class). If this field is not used, part 1 is assumed.</p> <p><i>ftype</i> A one-character field that indicates the file type. Valid values are:</p> <ul style="list-style-type: none"> <li><b>b</b> block special device</li> <li><b>c</b> character special device</li> <li><b>d</b> directory</li> <li><b>e</b> a file to be edited upon installation or removal (may be shared by several packages)</li> <li><b>f</b> a standard executable or data file</li> <li><b>i</b> installation script or information file</li> <li><b>l</b> linked file</li> <li><b>p</b> named pipe</li> <li><b>s</b> symbolic link</li> <li><b>v</b> volatile file (one whose contents are expected to change, like a log file)</li> <li><b>x</b> an exclusive directory accessible only by this package</li> </ul> <p><i>class</i> The installation class to which the file belongs. This name must contain only alphanumeric characters and be no longer than 12 characters. The field is not specified for installation scripts. (<b>admin</b> and all classes beginning with capital letters are reserved class names.)</p> <p><i>pathname</i> The pathname where the file will reside on the target machine, for example, <b>/usr/bin/mail</b> or <b>bin/ras/proc</b>. Relative pathnames (those that do not begin with a slash) indicate that the file is relocatable. The form</p> <p style="text-align: center;"><i>path1=path2</i></p> <p>may be used for two purposes: to define a link and to define local pathnames. For linked files, <i>path1</i> indicates the destination of the link and <i>path2</i> indicates the source file. (This format is mandatory for linked files.)</p> <p>For local pathnames, <i>path1</i> indicates the pathname an object should have on the machine where the entry is to be installed and <i>path2</i> indicates either a rela-</p>

	<p>tive or fixed pathname to a file on the host machine which contains the actual contents.</p> <p>A pathname may contain a variable specification of the form <i>\$variable</i>. If <i>variable</i> begins with a lower case letter, it is a build variable. If <i>variable</i> begins with an upper case letter, it is an install variable. Build variables are bound at build time. If an install variable is known at build time, its definition is inserted into the <b>pkginfo(4)</b> file so that it will be available at install time. If an install variable is not known at build time, it will be bound at install time.</p>
<i>major</i>	The major device number. The field is only specified for block or character special devices.
<i>minor</i>	The minor device number. The field is only specified for block or character special devices.
<i>mode</i>	<p>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 files or packaging information files.</p> <p>The mode can be a variable specification of the form <i>\$variable</i>. If <i>variable</i> begins with a lower case letter, it is a build variable. If <i>variable</i> begins with an upper case letter, it is an install variable. Build variables are bound at build time. If an install variable is known at build time, its definition is inserted into the <b>pkginfo(4)</b> file so that it will be available at install time. If an install variable is not known at build time, it will be bound at install time.</p>
<i>owner</i>	<p>The owner of the file (for example, <b>bin</b> or <b>root</b>). 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 files or packaging information files.</p> <p>The owner can be a variable specification of the form <i>\$variable</i>. If <i>variable</i> begins with a lower case letter, it is a build variable. If <i>variable</i> begins with an upper case letter, it is an install variable. Build variables are bound at build time. If an install variable is known at build time, its definition is inserted into the <b>pkginfo(4)</b> file so that it will be available at install time. If an install variable is not known at build time, it will be bound at install time.</p>
<i>group</i>	<p>The group to which the file belongs (for example, <b>bin</b> or <b>sys</b>). 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 files or packaging information files.</p> <p>The group can be a variable specification of the form <i>\$variable</i>. If <i>variable</i> begins with a lower case letter, it is a build variable. If <i>variable</i> begins with an upper case letter, it is an install variable. Build variables are bound at build time. If an install variable is known at build time, its definition is inserted into the <b>pkginfo(4)</b> file so that it will be available at install time. If an install variable is not known at build time, it will be bound at install time.</p>

An exclamation point (!) at the beginning of a line indicates that the line contains a command. These commands are used to incorporate files in other directories, to locate objects on a host machine, and to set permanent defaults. The following commands are available:

<b>search</b>	Specifies a list of directories (separated by white space) to search for when looking for file contents on the host machine. The base name of the <i>path</i> field is appended to each directory in the ordered list until the file is located. Searches are not recursive.
<b>include</b>	Specifies a pathname which points to another prototype file to include. Note that <b>search</b> requests do not span <b>include</b> files.
<b>default</b>	Specifies a list of attributes (mode, owner, and group) to be used by default if attribute information is not provided for prototype entries which require the information. The defaults do not apply to entries in <b>include</b> prototype files.
<i>param=value</i>	Places the indicated parameter in the current environment. Spans to subsequent included prototype files.

The above commands may have variable substitutions embedded within them, as demonstrated in the two example prototype files below.

Before files are overwritten during installation, they are copied to a temporary pathname. The exception to this rule is files whose mode includes execute permission, unless the file is editable (that is, *f*type is *e*). For files which meet this exception, the existing version is linked to a temporary pathname, and the original file is removed. This allows processes which are executing during installation to be overwritten.

## EXAMPLES

Example 1:

```

!PROJDIR=/usr/proj
!BIN=$PROJDIR/bin
!CFG=$PROJDIR/cfg
!LIB=$PROJDIR/lib
!HDRS=$PROJDIR/hdrs
!search /usr/myname/usr/bin /usr/myname/src /usr/myname/hdrs
i pkginfo=/usr/myname/wrap/pkginfo
i depend=/usr/myname/wrap/depend
i version=/usr/myname/wrap/version
d none /usr/wrap 0755 root bin
d none /usr/wrap/usr/bin 0755 root bin
! search $BIN
f none /usr/wrap/bin/INSTALL 0755 root bin
f none /usr/wrap/bin/REMOVE 0755 root bin
f none /usr/wrap/bin/addpkg 0755 root bin
!default 755 root bin
f none /usr/wrap/bin/audit
f none /usr/wrap/bin/listpkg

```

```

f none /usr/wrap/bin/pkgmk
# the following file starts out zero length but grows
v none /usr/wrap/logfile=/dev/null 0644 root bin
# the following specifies a link (dest=src)
l none /usr/wrap/src/addpkg=/usr/wrap/bin/rmpkg
! search $SRC
!default 644 root other
f src /usr/wrap/src/INSTALL.sh
f src /usr/wrap/src/REMOVE.sh
f src /usr/wrap/src/addpkg.c
f src /usr/wrap/src/audit.c
f src /usr/wrap/src/listpkg.c
f src /usr/wrap/src/pkgmk.c
d none /usr/wrap/data 0755 root bin
d none /usr/wrap/save 0755 root bin
d none /usr/wrap/spool 0755 root bin
d none /usr/wrap/tmp 0755 root bin
d src /usr/wrap/src 0755 root bin

```

Example 2:

```

# this prototype is generated by 'pkgproto' to refer
# to all prototypes in my src directory
!PROJDIR=/usr/dew/projx
!include $PROJDIR/src/cmd/prototype
!include $PROJDIR/src/cmd/audmerg/protofile
!include $PROJDIR/src/lib/proto

```

**SEE ALSO** [pkgmk\(1\)](#), [pkginfo\(4\)](#)

*Application Packaging Developer's Guide*

**NOTES**

Normally, if a file is defined in the **prototype** file but does not exist, that file is created at the time of package installation. However, if the file pathname includes a directory that does not exist, the file will not be created. For example, if the **prototype** file has the following entry:

```
f none /usr/dev/bin/command
```

and that file does not exist, it will be created if the directory **/usr/dev/bin** already exists or if the **prototype** also has an entry defining the directory:

```
d none /usr/dev/bin
```



<b>NAME</b>	pseudo – configuration files for pseudo device drivers
<b>DESCRIPTION</b>	<p>Pseudo devices are devices that are implemented entirely in software. Drivers for pseudo devices must provide driver configuration files to inform the system of each pseudo device that should be created.</p> <p>Configuration files for pseudo device drivers must identify the parent driver explicitly as <i>pseudo</i>, and must create an integer property called <i>instance</i> which is unique to this entry in the configuration file.</p> <p>Each entry in the configuration file creates a prototype devinfo node. Each node is assigned an instance number which is determined by the value of the <i>instance</i> property. This property is only applicable to children of the <i>pseudo</i> parent, and is required since pseudo devices have no hardware address from which to determine the instance number. See <b>driver.conf(4)</b> for further details of configuration file syntax.</p>
<b>EXAMPLES</b>	<p>Here is a configuration file called <b>ramdisk.conf</b> for a pseudo device driver that implements a RAM disk. This file creates two nodes called "ramdisk". The first entry creates ramdisk node instance 0, and the second creates ramdisk node, instance 1, with the additional disk-size property set to 512.</p> <pre># # Copyright (c) 1993, by Sun Microsystems, Inc. # #ident "@(#)ramdisk.conf 1.3 93/06/04 SMI"  name="ramdisk" parent="pseudo" instance=0; name="ramdisk" parent="pseudo" instance=1 disk-size=512;</pre>
<b>SEE ALSO</b>	<p><b>driver.conf(4)</b>, <b>ddi_prop_op(9F)</b></p> <p><i>Writing Device Drivers</i></p>

<b>NAME</b>	publickey – public key database
<b>SYNOPSIS</b>	<b>/etc/publickey</b>
<b>DESCRIPTION</b>	<p><b>/etc/publickey</b> is a local public key database that is used for secure RPC. The <b>/etc/publickey</b> file can be used in conjunction with or instead of other publickey databases, including the NIS publickey map and the NIS+ publickey map. Each entry in the database consists of a network user name (which may refer to either a user or a host-name), followed by the user's public key (in hex notation), a colon, and then the user's secret key encrypted with a password (also in hex notation).</p> <p>The <b>/etc/publickey</b> file contains a default entry for <b>nobody</b>.</p>
<b>SEE ALSO</b>	<b>chkey(1)</b> , <b>newkey(1M)</b> , <b>getpublickey(3N)</b> , <b>nsswitch.conf(4)</b>

<b>NAME</b>	queuedefs – queue description file for at, batch, and cron
<b>SYNOPSIS</b>	<code>/etc/cron.d/queuedefs</code>
<b>DESCRIPTION</b>	<p>The <b>queuedefs</b> file describes the characteristics of the queues managed by <b>cron(1M)</b>. Each non-comment line in this file describes one queue. The format of the lines are as follows:</p> <pre style="margin-left: 40px;">q.[njob][nicen][nwaitw]</pre> <p>The fields in this line are:</p> <p><i>q</i>        The name of the queue. <b>a</b> is the default queue for jobs started by <b>at(1)</b>; <b>b</b> is the default queue for jobs started by <b>batch</b> (see <b>at(1)</b>); <b>c</b> is the default queue for jobs run from a <b>crontab(1)</b> file.</p> <p><i>njob</i>     The maximum number of jobs that can be run simultaneously in that queue; if more than <i>njob</i> jobs are ready to run, only the first <i>njob</i> jobs will be run, and the others will be run as jobs that are currently running terminate. The default value is <b>100</b>.</p> <p><i>nice</i>     The <b>nice(1)</b> value to give to all jobs in that queue that are not run with a user ID of super-user. The default value is <b>2</b>.</p> <p><i>nwait</i>    The number of seconds to wait before rescheduling a job that was deferred because more than <i>njob</i> jobs were running in that job's queue, or because the system-wide limit of jobs executing has been reached. The default value is <b>60</b>.</p> <p>Lines beginning with # are comments, and are ignored.</p>
<b>EXAMPLES</b>	<pre style="margin-left: 40px;"># # a.4j1n b.2j2n90w</pre> <p>This file specifies that the <b>a</b> queue, for <b>at</b> jobs, can have up to 4 jobs running simultaneously; those jobs will be run with a <b>nice</b> value of 1. As no <i>nwait</i> value was given, if a job cannot be run because too many other jobs are running <b>cron</b> will wait 60 seconds before trying again to run it.</p> <p>The <b>b</b> queue, for <b>batch(1)</b> jobs, can have up to 2 jobs running simultaneously; those jobs will be run with a <b>nice(1)</b> value of 2. If a job cannot be run because too many other jobs are running, <b>cron(1M)</b> will wait 90 seconds before trying again to run it. All other queues can have up to 100 jobs running simultaneously; they will be run with a <b>nice</b> value of 2, and if a job cannot be run because too many other jobs are running <b>cron</b> will wait 60 seconds before trying again to run it.</p>
<b>FILES</b>	<code>/etc/cron.d/queuedefs</code> queue description file for <b>at</b> , <b>batch</b> , and <b>cron</b> .

**SEE ALSO**

**at(1), crontab(1), nice(1), cron(1M)**

<b>NAME</b>	remote – remote host description file																						
<b>SYNOPSIS</b>	<code>/etc/remote</code>																						
<b>DESCRIPTION</b>	<p>The systems known by <b>tip</b>(1) and their attributes are stored in an ASCII file which is structured somewhat like the <b>termcap</b> file. Each line in the file provides a description for a single <i>system</i>. Fields are separated by a colon ':'. Lines ending in a '\ ' character with an immediately following NEWLINE are continued on the next line.</p> <p>The first entry is the name(s) of the host system. If there is more than one name for a system, the names are separated by vertical bars. After the name of the system comes the fields of the description. A field name followed by an '=' sign indicates a string value follows. A field name followed by a '#' sign indicates a following numeric value.</p> <p>Entries named <b>tipbaudrate</b> are used as default entries by <b>tip</b>, as follows. When <b>tip</b> is invoked with only a phone number, it looks for an entry of the form <b>tipbaudrate</b>, where <i>baudrate</i> is the baud rate with which the connection is to be made. For example, if the connection is to be made at <b>300</b> baud, <b>tip</b> looks for an entry of the form <b>tip300</b>.</p>																						
<b>CAPABILITIES</b>	<p>Capabilities are either strings (<b>str</b>), numbers (<b>num</b>), or boolean flags (<b>bool</b>). A string capability is specified by <i>capability=value</i>; for example, '<b>dv=/dev/harris</b>'. A numeric capability is specified by <i>capability#value</i>; for example, '<b>xa#99</b>'. A boolean capability is specified by simply listing the capability.</p> <p><b>at</b> (<b>str</b>) Auto call unit type. The following lists valid 'at' types and their corresponding hardware:</p> <table border="0"> <tr><td><b>biz31f</b></td><td>Bizcomp 1031, tone dialing</td></tr> <tr><td><b>biz31w</b></td><td>Bizcomp 1031, pulse dialing</td></tr> <tr><td><b>biz22f</b></td><td>Bizcomp 1022, tone dialing</td></tr> <tr><td><b>biz22w</b></td><td>Bizcomp 1022, pulse dialing</td></tr> <tr><td><b>df02</b></td><td>DEC DF02</td></tr> <tr><td><b>df03</b></td><td>DEC DF03</td></tr> <tr><td><b>ventel</b></td><td>Ventel 212+</td></tr> <tr><td><b>v3451</b></td><td>Vadic 3451 Modem</td></tr> <tr><td><b>v831</b></td><td>Vadic 831</td></tr> <tr><td><b>hayes</b></td><td>Any Hayes-compatible modem</td></tr> <tr><td><b>at</b></td><td>Any Hayes-compatible modem</td></tr> </table> <p><b>br</b> (<b>num</b>) The baud rate used in establishing a connection to the remote host. This is a decimal number. The default baud rate is <b>300</b> baud.</p> <p><b>cm</b> (<b>str</b>) An initial connection message to be sent to the remote host. For example, if a host is reached through a port selector, this might be set to the appropriate sequence required to switch to the host.</p> <p><b>cu</b> (<b>str</b>) Call unit if making a phone call. Default is the same as the <b>dv</b> field.</p> <p><b>db</b> (<b>bool</b>) Cause <b>tip</b>(1) to ignore the first hangup it sees. <b>db</b> (dialback) allows the user to remain in <b>tip</b> while the remote machine disconnects and places a call back to the local machine. For more information about dialback configuration, see</p>	<b>biz31f</b>	Bizcomp 1031, tone dialing	<b>biz31w</b>	Bizcomp 1031, pulse dialing	<b>biz22f</b>	Bizcomp 1022, tone dialing	<b>biz22w</b>	Bizcomp 1022, pulse dialing	<b>df02</b>	DEC DF02	<b>df03</b>	DEC DF03	<b>ventel</b>	Ventel 212+	<b>v3451</b>	Vadic 3451 Modem	<b>v831</b>	Vadic 831	<b>hayes</b>	Any Hayes-compatible modem	<b>at</b>	Any Hayes-compatible modem
<b>biz31f</b>	Bizcomp 1031, tone dialing																						
<b>biz31w</b>	Bizcomp 1031, pulse dialing																						
<b>biz22f</b>	Bizcomp 1022, tone dialing																						
<b>biz22w</b>	Bizcomp 1022, pulse dialing																						
<b>df02</b>	DEC DF02																						
<b>df03</b>	DEC DF03																						
<b>ventel</b>	Ventel 212+																						
<b>v3451</b>	Vadic 3451 Modem																						
<b>v831</b>	Vadic 831																						
<b>hayes</b>	Any Hayes-compatible modem																						
<b>at</b>	Any Hayes-compatible modem																						

*TCP/IP and Data Communications Administration Guide.*

- di** (str) Disconnect message sent to the host when a disconnect is requested by the user.
- du** (bool) This host is on a dial-up line.
- dv** (str) Device(s) to open to establish a connection. If this file refers to a terminal line, **tip** attempts to perform an exclusive open on the device to insure only one user at a time has access to the port.
- ec** (bool) Initialize the **tip** variable **echocheck** to *on*, so that **tip** will synchronize with the remote host during file transfer by waiting for the echo of the last character transmitted.
- el** (str) Characters marking an end-of-line. The default is no characters. **tip** only recognizes “~” escapes after one of the characters in **el**, or after a RETURN.
- es** (str) The command prefix (escape) character for **tip**.
- et** (num) Number of seconds to wait for an echo response when echo-check mode is on. This is a decimal number. The default value is 10 seconds.
- ex** (str) Set of non-printable characters not to be discarded when scripting with beautification turned on. The default value is “\t\n\b\f”.
- fo** (str) Character used to force literal data transmission. The default value is ‘\377’.
- fs** (num) Frame size for transfers. The default frame size is equal to 1024.
- hd** (bool) Initialize the **tip** variable **halfduplex** to *on*, so local echo should be performed.
- hf** (bool) Initialize the **tip** variable **hardwareflow** to *on*, so hardware flow control is used.
- ie** (str) Input end-of-file marks. The default is a null string (“”).
- nb** (bool) Initialize the **tip** variable **beautify** to *off*, so that unprintable characters will not be discarded when scripting.
- nt** (bool) Initialize the **tip** variable **tandem** to *off*, so that XON/XOFF flow control will not be used to throttle data from the remote host.
- nv** (bool) Initialize the **tip** variable **verbose** to *off*, so that verbose mode will be turned on.
- oe** (str) Output end-of-file string. The default is a null string (“”). When **tip** is transferring a file, this string is sent at end-of-file.
- pa** (str) The type of parity to use when sending data to the host. This may be one of **even**, **odd**, **none**, **zero** (always set bit 8 to 0), **one** (always set bit 8 to 1). The default is **none**.
- pn** (str) Telephone number(s) for this host. If the telephone number field contains an ‘@’ sign, **tip** searches the **/etc/phones** file for a list of telephone numbers — see **phones(4)**. A ‘%’ sign in the telephone number indicates a 5-second delay for the Ventel Modem.

For Hayes-compatible modems, if the telephone number starts with an 'S', the telephone number string will be sent to the modem without the "DT", which allows reconfiguration of the modem's S-registers and other parameters; for example, to disable auto-answer: "**pn=S0=0DT5551234**"; or to also restrict the modem to return only the basic result codes: "**pn=S0=0X0DT5551234**".

- pr** (str) Character that indicates end-of-line on the remote host. The default value is '\n'.
- ra** (bool) Initialize the **tip** variable **raise** to *on*, so that lower case letters are mapped to upper case before sending them to the remote host.
- rc** (str) Character that toggles case-mapping mode. The default value is '\377'.
- re** (str) The file in which to record session scripts. The default value is **tip.record**.
- rw** (bool) Initialize the **tip** variable **rawftp** to *on*, so that all characters will be sent as is during file transfers.
- sc** (bool) Initialize the **tip** variable **script** to *on*, so that everything transmitted by the remote host will be recorded.
- tb** (bool) Initialize the **tip** variable **tabexpand** to *on*, so that tabs will be expanded to spaces during file transfers.
- tc** (str) Indicates that the list of capabilities is continued in the named description. This is used primarily to share common capability information.

**EXAMPLES**

Here is a short example showing the use of the capability continuation feature:

```
UNIX-1200:\
:dv=/dev/cua0:el=^D^U^C^S^Q^O@:du:at=ventel:ie=#$%:oe=^D:br#1200:
arpavax | ax:\
:pn=7654321%:tc=UNIX-1200
```

**FILES**

**/etc/remote** remote host description file.  
**/etc/phones** remote host phone number database.

**SEE ALSO**

**tip(1)**, **phones(4)**

*TCP/IP and Data Communications Administration Guide*

<b>NAME</b>	resolv.conf – configuration file for name server routines
<b>DESCRIPTION</b>	<p>This file helps initialize routines from the <b>resolver(3N)</b> C library. The resolver routines provide access to the Internet Domain Name System.</p> <p>The resolver configuration file contains information that is read by the resolver routines the first time a process calls them. The file is designed to be human readable and contains a list of keyword-value pairs that provide various types of resolver information. Keyword-value pairs are of the form:</p> <p style="text-align: center;"><i>keyword value</i></p> <p>The different configuration options are:</p> <p><b>nameserver address</b> Specifies the Internet address in dot-notation format of one name server to which the resolver should direct any queries. Up to <i>MAXNS</i> (currently three) name servers may be listed, on as many as <i>MAXNS</i> <b>nameserver</b> lines in <b>resolv.conf</b>. If multiple servers are specified, the resolver routines query them in the order listed. If no <b>nameserver</b> lines are present in the file, resolver routines use the name server on the local machine.</p> <p>The algorithm of the resolver routines is: try the first name server specified. If the query times out, try the next server listed in the configuration file, and so on until the complement of servers there has been exhausted. If those queries also time out, try the full complement of name servers again, until the maximum number of retry passes has been made.</p> <p><b>domain name</b> Specifies a local domain name for use as the default domain.</p> <p>Most queries for names within a domain can use short names relative to the local domain. If a <b>domain</b> line is missing from the configuration file, the domain is determined from the environment variable, <b>LOCALDOMAIN</b>, if it is defined, from the domain name (see <b>domainname(1M)</b>) by omitting the first level, or from the host name (<b>gethostname(3C)</b>) by using everything after the first dot. Finally, if the host name does not contain a domain part, the root domain is assumed.</p> <p><b>search searchlist</b> Specifies a search list for host-name lookup. The search list is normally determined from the local domain name; by default, it contains only the local domain name. This may be changed by listing the desired domains for searches in <i>searchlist</i>. Spaces or tabs must separate domain names.</p> <p>Most resolver queries are attempted using each component of the search path in turn until a match is found. Note that this process may be slow and will generate a lot of network traffic if the servers for the listed domains are not local. Also queries will time out if no server is available for one of the domains.</p>



<b>sortlist</b> <i>addresslist</i>	<p>The search list is currently limited to six domains with a total of 256 characters.</p> <p>Causes addresses returned by <b>gethostbyname(3C)</b> to be sorted in accordance with local rules. A sortlist is specified by IP address netmask pairs. The netmask is optional and defaults to the natural netmask of the net. The IP address and optional network pairs are separated by slashes. Up to 10 pairs may be specified. For example, the following specification requires <b>gethostbyname()</b> to return the netmask pair <b>130.155.160.0/255.255.240.0</b> ahead of the IP address <b>130.155.0.0</b>.</p> <p style="text-align: center;"><b>sortlist 130.155.160.0/255.255.240.0 130.155.0.0</b></p>
<b>options</b> <i>optionlist</i>	<p>Specifies optional behaviors for various resolver routines in accordance with <i>optionlist</i> values, each of which is equivalent to an internal resolver variable.</p> <p>The values that may be included as individual <i>optionlist</i> values are:</p> <p><b>debug</b> Sets <b>RES_DEBUG</b> in the <b>_res.options</b> field.</p> <p><b>ndots:n</b> Sets a floor threshold for the number of dots which must appear in a name given to <b>res_query()</b> (see <b>resolver(3N)</b>) before an initial absolute (as-is) query is performed. The default for <i>n</i> is <b>1</b>. Thus, if there are any dots in a name, the name is tried first as an absolute name before any search-list domain names are appended to it.</p> <p><b>retry:n</b> Sets the number of attempts made to connect to each name server. While <b>retry:0</b> is allowed, it is equivalent to <b>retry:1</b>. The default is 4.</p> <p><b>retrans:n</b> Sets the basic retransmit timeout, in seconds. The default is 5. An exponential backoff algorithm is used, so the default values for <b>retry</b> and <b>retrans</b> result in <math>5+10+20+40=75</math> seconds of total timeout for each name server. While <b>retrans:0</b> is allowed, it is equivalent to <b>retrans:1</b>.</p> <p>The <b>domain</b> and <b>search</b> keywords are mutually exclusive. If more than one instance of these keywords is present, the last instance takes precedence.</p> <p>The options established through any <b>search</b> lines in the local <b>resolv.conf</b> file can be overridden on a per-process basis by setting the environment variable, <b>LOCALDOMAIN</b>, to a space-separated list of search domains.</p> <p>The options established through any <b>options</b> lines in the local <b>resolv.conf</b> file can be amended on a per-process basis by setting the environment variable, <b>RES_OPTIONS</b>, to a space-separated list of resolver options. These options are listed above under the <b>options</b> keyword.</p>

The keyword-value pair must appear on a single line, and the keyword (for instance, **nameserver**) must start the line. The value or value list follows the keyword, separated from it by white space characters.

**FILES** /etc/resolv.conf

**SEE ALSO** **domainname(1M)**, **in.named(1M)**, **gethostbyname(3N)**, **gethostbyname(3C)**, **resolver(3N)**  
Vixie, Paul;Dunlap, Keven J., Karels, Michael J., *Name Server Operations Guide for BIND* (public domain), Internet Software Consortium, 1996.

<b>NAME</b>	rmmount.conf – removable media mounter configuration file
<b>SYNOPSIS</b>	<i>/etc/rmmount.conf</i>
<b>DESCRIPTION</b>	<p>The <b>rmmount.conf</b> file contains the <b>rmmount(1M)</b> configuration information. This file describes where to find shared objects that perform actions on file systems after identifying and mounting them. The <b>rmmount.conf</b> file is also used to share CD-ROM and floppy file systems.</p> <p>Actions are executed in the order in which they appear in the configuration file. The action function can return either <b>1</b> or <b>0</b>. If it returns <b>0</b>, no further actions will be executed. This allows the function to control which applications are executed. For example, <b>action_filemgr</b> always returns <b>0</b> if the File Manager is running, thereby preventing subsequent actions from being executed.</p> <p>To execute an action after media has been inserted and while the File Manager is not running, list the action after <b>action_filemgr</b> in the <b>rmmount.conf</b> file. To execute an action before the File Manager becomes aware of the media, list the action before <b>action_filemgr</b> in the <b>rmmount.conf</b> file.</p> <p>The syntax for the <b>rmmount.conf</b> file is as follows.</p> <pre> # File system identification ident filesystem_type shared_object media_type [media_type ... ] # Actions action media_type shared_object args_to_so # File system sharing share media_or_file_system share_command_options # Mount command options mount media_or_file_system [file_system_spec] -o mount_command_options </pre> <p>Explanations of the syntax for the <b>File system identification</b> fields are as follows.</p> <p><i>filesystem_type</i> An ASCII string used as the file system type flag of the <b>mount</b> command (see the <b>-F</b> option of <b>mount(1M)</b>). It is also used to match names passed to <b>rmmount(1M)</b> from Volume Management.</p> <p><i>shared_object</i> Programs that identify file systems and perform actions. This <i>shared_object</i> is found at <b>/usr/lib/fs/ filesystem_type/ shared_object</b>.</p> <p><i>media_type</i> The type of media where this file system resides. Legal values are <b>cdrom</b> and <b>floppy</b>.</p> <p>Explanations of the syntax for the <b>Actions</b> fields are as follows.</p> <p><i>media_type</i> Type of media. This argument is passed in from Volume Management as <b>VOLUME_TYPE</b>.</p> <p><i>shared_object</i> Programs that identify file systems and perform actions. If <i>shared_object</i> starts with '/' (slash), the full path name is used; otherwise, <b>/usr/lib/rmmount</b> is prepended to the name.</p>

*args\_to\_so* Arguments passed to the *shared\_object*. These arguments are passed in as an *argc* and *argv[]*.

The definition of the interface to **Actions** is located in **/usr/include/rmmount.h**.

Explanations of the syntax for the **File system sharing** fields are as follows.

*media\_or\_file\_system*

Either the type of media (CD-ROM or floppy) or the specific file system to share.

*share\_command\_options*

Options of the **share** command. See **share(1M)** for more information about these options.

Explanations of the syntax for the **Mount command options** fields are as follows.

*media\_or\_file\_system*

Either the type of media (CD-ROM or floppy) or the specific file system to share.

*file\_system\_spec* Specifies one or more file systems to which this line applies. Defaults to "all" filesystem types.

*mount\_command\_options*

One or more options to be passed to the **mount** command.

#### Default Values

The following is an example of an **rmmount.conf** file.

```
#
# Removable Media Mounter configuration file.
#
# File system identification
ident hsfs ident_hsfs.so cdrom
ident ufs ident_ufs.so cdrom floppy
ident pcfs ident_pcfs.so floppy
# Actions
action cdrom action_filemgr.so
action floppy action_filemgr.so
```

#### EXAMPLES

The following examples show how various file systems are shared using the share syntax for the **rmmount.conf** file. These lines are added after the Actions entries.

**share cdrom\*** Shares all CD-ROMs via NFS and applies no access restrictions.

**share solaris\_2.x\***  
Shares CD-ROMs named *solaris\_2.x\** with no access restrictions.

**share cdrom\* -o ro=engineering**  
Shares all CD-ROMs via NFS but exports only to the "engineering" net-group.

**share solaris\_2.x\* -d distribution CD**  
Shares CD-ROMs named *solaris\_2.x\** with no access restrictions and

with the description that it is a distribution CD-ROM.

**share floppy0** Shares any floppy inserted into floppy drive 0.

The following examples show how different **mount** options could be used to customize how **rmmount** mounts media:

**mount cdrom\* hsfs -o norr**

mounts all High Sierra CD-ROMs with the "norr" (no Rock Ridge extensions) option (see **mount\_hfsfs(1M)**)

**mount floppy1 -o ro**

will always mount the second floppy disk read-only (for all filesystem types)

**SEE ALSO**

**volcancel(1)**, **volcheck(1)**, **volmissing(1)**, **mount(1M)**, **mount\_hfsfs(1M)**, **rmmount(1M)**, **share(1M)**, **vold(1M)**, **vold.conf(4)**, **volfs(7FS)**

**NOTES**

The **mount** command will fail if you specify an incorrect mount option and filesystem combination. When using the "mount" options line, ensure that the options specified will work with the file system types specified.

<b>NAME</b>	rmtab – remote mounted file system table
<b>SYNOPSIS</b>	<b>/etc/rmtab</b>
<b>DESCRIPTION</b>	<p><b>rmtab</b> contains a table of filesystems that are remotely mounted by NFS clients. This file is maintained by <b>mountd</b>(1M), the mount daemon. The data in this file should be obtained only from <b>mountd</b>(1M) using the <b>MOUNTPROC_DUMP</b> remote procedure call.</p> <p>The file contains a line of information for each remotely mounted filesystem. There are a number of lines of the form:</p> <p style="text-align: center;"><i>hostname:fsname</i></p> <p>The mount daemon adds an entry for any client that successfully executes a mount request and deletes the appropriate entries for an unmount request.</p> <p>Lines beginning with a hash ( '#') are commented out. These lines are removed from the file by <b>mountd</b>(1M) when it first starts up. Stale entries may accumulate for clients that crash without sending an unmount request.</p>
<b>FILES</b>	<b>/etc/rmtab</b>
<b>SEE ALSO</b>	<b>mountd</b> (1M), <b>showmount</b> (1M)

<b>NAME</b>	route – kernel packet forwarding database
<b>SYNOPSIS</b>	<pre>#include &lt;sys/types.h&gt; #include &lt;sys/socket.h&gt; #include &lt;net/if.h&gt; #include &lt;net/route.h&gt;  int socket(PF_ROUTE, SOCK_RAW, AF_INET);</pre>
<b>DESCRIPTION</b>	<p>UNIX provides some packet routing facilities. The kernel maintains a routing information database, which is used in selecting the appropriate network interface when transmitting packets.</p> <p>A user process (or possibly multiple co-operating processes) maintains this database by sending messages over a special kind of socket. This supplants fixed size <code>ioctl(2)</code>'S specified in <code>routing(4)</code>. Routing table changes may only be carried out by the super user. The operating system may spontaneously emit routing messages in response to external events, such as receipt of a re-direct, or failure to locate a suitable route for a request. The message types are described in greater detail below.</p> <p>Routing database entries come in two flavors: entries for a specific host, or entries for all hosts on a generic subnetwork (as specified by a bit mask and value under the mask). The effect of wildcard or default route may be achieved by using a mask of all zeros, and there may be hierarchical routes.</p> <p>When the system is booted and addresses are assigned to the network interfaces, the internet protocol family installs a routing table entry for each interface when it is ready for traffic. Normally the protocol specifies the route through each interface as a <i>direct</i> connection to the destination host or network. If the route is direct, the transport layer of a protocol family usually requests the packet be sent to the same host specified in the packet. Otherwise, the interface is requested to address the packet to the gateway listed in the routing entry (i.e., the packet is forwarded).</p> <p>When routing a packet, the kernel attempts to find the most specific route matching the destination. If no entry is found, the destination is declared to be unreachable, and a routing-miss message is generated if there are any listeners on the routing control socket (described below). If there are two different mask and value-under-the-mask pairs that match, the more specific is the one with more bits in the mask. A route to a host is regarded as being supplied with a mask of as many ones as there are bits in the destination.</p> <p>Note: a wildcard routing entry is specified with a zero destination address value, and a mask of all zeroes. Wildcard routes are used when the system fails to find other routes matching the destination. The combination of wildcard routes and routing redirects can provide an economical mechanism for routing traffic.</p> <p>One opens the channel for passing routing control messages by using the socket call shown in the <b>SYNOPSIS</b> section above. There can be more than one routing socket open per system.</p>

Messages are formed by a header followed by a small number of sockadders, whose length depend on the address family. Sockadders are interpreted by position. An example of a type of message with three addresses might be a CIDR prefix route: Destination, Net-mask, and Gateway. The interpretation of which addresses are present is given by a bit mask within the header, and the sequence is least significant to most significant bit within the vector.

Any messages sent to the kernel are returned, and copies are sent to all interested listeners. The kernel provides the process ID of the sender, and the sender may use an additional sequence field to distinguish between outstanding messages. However, message replies may be lost when kernel buffers are exhausted.

The kernel may reject certain messages, and will indicate this by filling in the `rtm_errno` field of the `rt_msghdr` struct (see below). The following codes may be returned:

<b>EEXIST</b>	If requested to duplicate an existing entry
<b>ESRCH</b>	If requested to delete a non-existent entry
<b>ENOBUFS</b>	If insufficient resources were available to install a new route.

In the current implementation, all routing processes run locally, and the values for `rtm_errno` are available through the normal `errno` mechanism, even if the routing reply message is lost.

A process may avoid the expense of reading replies to its own messages by issuing a `setsockopt(3N)` call indicating that the `SO_USELOOPBACK` option at the `SOL_SOCKET` level is to be turned off. A process may ignore all messages from the routing socket by doing a `shutdown(3N)` system call for further input.

If a route is in use when it is deleted, the routing entry is marked down and removed from the routing table, but the resources associated with it are not reclaimed until all references to it are released.

#### Messages

User processes can obtain information about the routing entry to a specific destination by using a `RTM_GET` message.

Messages include:

```
#define RTM_ADD           /* Add Route */
#define RTM_DELETE       /* Delete Route */
#define RTM_CHANGE       /* Change Metrics, Flags, or Gateway */
#define RTM_GET          /* Report Information */
#define RTM_LOOSING      /* Kernel Suspects Partitioning */
#define RTM_REDIRECT     /* Told to use different route */
#define RTM_MISS         /* Lookup failed on this address */
#define RTM_RESOLVE      /* request to resolve dst to LL addr */
#define RTM_NEWADDR      /* address being added to iface */
#define RTM_DELADDR      /* address being removed from iface */
#define RTM_IFINFO       /* iface going up/down etc. */
```



A message header consists of:

```

struct rt_msghdr {
  u_short rtm_msglen;      /* to skip over non-understood messages */
  u_char  rtm_version;    /* future binary compatibility */
  u_char  rtm_type;       /* message type */
  u_short rtm_index;      /* index for associated ifp */
  pid_t   rtm_pid;        /* identify sender */
  int     rtm_addrs;      /* bitmask identifying sockaddrs in msg */
  int     rtm_seq;        /* for sender to identify action */
  int     rtm_errno;      /* why failed */
  int     rtm_flags;      /* flags, incl kern & message, e.g., DONE */
  int     rtm_use;        /* from rtenry */
  u_long  rtm_inits;      /* which values we are initializing */
  struct  rt_metrics rtm_rmx; /* metrics themselves */
};

```

where,

```

struct rt_metrics {
  u_long rmx_locks;      /* Kernel must leave these values alone */
  u_long rmx_mtu;        /* MTU for this path */
  u_long rmx_hopcount;   /* max hops expected */
  u_long rmx_expire;     /* lifetime for route, e.g., redirect */
  u_long rmx_recvpipe;   /* inbound delay-bandwidth product */
  u_long rmx_sendpipe;   /* outbound delay-bandwidth product */
  u_long rmx_ssthresh;   /* outbound gateway buffer limit */
  u_long rmx_rtt;        /* estimated round trip time */
  u_long rmx_rttvar;     /* estimated rtt variance */
  u_long rmx_pksent;     /* packets sent using this route */
};

```

Flags include the values:

```

#define RTF_UP          /* route usable */
#define RTF_GATEWAY    /* destination is a gateway */
#define RTF_HOST       /* host entry (net otherwise) */
#define RTF_REJECT     /* host or net unreachable */
#define RTF_DYNAMIC    /* created dynamically (by redirect) */
#define RTF_MODIFIED   /* modified dynamically (by redirect) */
#define RTF_DONE       /* message confirmed */
#define RTF_MASK       /* subnet mask present */
#define RTF_CLONING    /* generate new routes on use */
#define RTF_XRESOLVE   /* external daemon resolves name */
#define RTF_LLINFO     /* generated by ARP */

```

```
#define RTF_STATIC      /* manually added */
#define RTF_BLACKHOLE  /* just discard pkts (during updates) */
#define RTF_PROTO1     /* protocol specific routing flag #1 */
#define RTF_PROTO2     /* protocol specific routing flag #2 */
```

Specifiers for metric values in `rmx_locks` and `rtm_inits` are:

```
#define RTV_MTU        /* init or lock _mtu */
#define RTV_HOPCOUNT /* init or lock _hopcount */
#define RTV_RPIPE      /* init or lock _recvpipe */
#define RTV_SPIPE      /* init or lock _sendpipe */
#define RTV_SSTHRESH   /* init or lock _ssthresh */
#define RTV_RTT        /* init or lock _rtt */
#define RTV_RTTVAR     /* init or lock _rttvar */
```

Specifiers for which addresses are present in the messages are:

```
#define RTA_DST        /* destination sockaddr present */
#define RTA_GATEWAY    /* gateway sockaddr present */
#define RTA_NETMASK    /* netmask sockaddr present */
#define RTA_GENMASK    /* cloning mask sockaddr present */
#define RTA_IFP        /* interface name sockaddr present */
#define RTA_IFA        /* interface addr sockaddr present */
#define RTA_AUTHOR     /* sockaddr for author of redirect */
#define RTA_BRD        /* for NEWADDR, broadcast or p-p dest addr */
```

**SEE ALSO** `ioctl(2)`, `setsockopt(3N)`, `shutdown(3N)`, `routing(4)`

**NOTES** Some of the metrics may not be implemented and return zero. The implemented metrics are set in `rtm_inits`.

<b>NAME</b>	routing – system support for packet network routing
<b>DESCRIPTION</b>	<p>The network facilities provide general packet routing. Routing table maintenance may be implemented in applications processes.</p> <p>A simple set of data structures compose a “routing table” used in selecting the appropriate network interface when transmitting packets. This table contains a single entry for each route to a specific network or host. The routing table was designed to support routing for the Internet Protocol (IP), but its implementation is protocol independent and thus it may serve other protocols as well. User programs may manipulate this data base with the aid of two <code>ioctl(2)</code> commands, <code>SIOCADDRT</code> and <code>SIOCDELRT</code>. These commands allow the addition and deletion of a single routing table entry, respectively. Routing table manipulations may only be carried out by privileged user.</p> <p>A routing table entry has the following form, as defined in <code>/usr/include/net/route.h</code>:</p> <pre> struct rtentry {     u_long          rt_hash;      /* to speed lookups */     struct sockaddr rt_dst;      /* key */     struct sockaddr rt_gateway;  /* value */     short          rt_flags;     /* up/down?, host/net */     short          rt_refcnt;    /* # held references */     u_long          rt_use;      /* raw # packets forwarded */ #ifdef STRNET     struct ip_provider *rt_prov; /* the answer: provider to use */ #else     struct ifnet      *rt_ifp;   /* the answer: interface to use */ #endif /* STRNET */ }; </pre> <p>with <code>rt_flags</code> defined from:</p> <pre> #define RTF_UP          0x1      /* route usable */ #define RTF_GATEWAY    0x2      /* destination is a gateway */ #define RTF_HOST       0x4      /* host entry (net otherwise) */ </pre> <p>Routing table entries come in three flavors: for a specific host, for all hosts on a specific network, for any destination not matched by entries of the first two types (a wildcard route). Each network interface installs a routing table entry when it is initialized. Normally the interface specifies the route through it is a “direct” connection to the destination host or network. If the route is direct, the transport layer of a protocol family usually requests the packet be sent to the same host specified in the packet. Otherwise, the interface may be requested to address the packet to an entity different from the eventual recipient (that is, the packet is forwarded).</p> <p>Routing table entries installed by a user process may not specify the hash, reference count, use, or interface fields; these are filled in by the routing routines. If a route is in use when it is deleted (<code>rt_refcnt</code> is non-zero), the resources associated with it will not be reclaimed until all references to it are removed.</p>

User processes read the routing tables through the **/dev/ip** device.

The *rt\_use* field contains the number of packets sent along the route. This value is used to select among multiple routes to the same destination. When multiple routes to the same destination exist, the least used route is selected.

A wildcard routing entry is specified with a zero destination address value. Wildcard routes are used only when the system fails to find a route to the destination host and network. The combination of wildcard routes and routing redirects can provide an economical mechanism for routing traffic.

<b>ERRORS</b>	<b>EEXIST</b>	A request was made to duplicate an existing entry.
	<b>ESRCH</b>	A request was made to delete a non-existent entry.
	<b>ENOBUFS</b>	Insufficient resources were available to install a new route.
	<b>ENOMEM</b>	Insufficient resources were available to install a new route.
	<b>ENETUNREACH</b>	The gateway is not directly reachable i.e. it does not match the destination/subnet on any of the network interfaces.
<b>FILES</b>	<b>/dev/ip</b>	IP device driver
<b>SEE ALSO</b>	<b>route(1M), ioctl(2)</b>	

<b>NAME</b>	rpc – rpc program number data base
<b>SYNOPSIS</b>	<code>/etc/rpc</code>
<b>DESCRIPTION</b>	<p>The <b>rpc</b> file is a local source containing user readable names that can be used in place of RPC program numbers. The <b>rpc</b> file can be used in conjunction with or instead of other rpc sources, including the NIS maps “rpc.byname” and “rpc.bynumber” and the NIS+ table “rpc”.</p> <p>The rpc file has one line for each RPC program name. The line has the following format:</p> <p style="text-align: center;"><i>name-of-the-RPC-program    RPC-program-number    aliases</i></p> <p>Items are separated by any number of blanks and/or tab characters. A “#” indicates the beginning of a comment; characters up to the end of the line are not interpreted by routines which search the file.</p>
<b>EXAMPLES</b>	<p>Below is an example of an RPC database:</p> <pre># #           rpc # rpcbind    100000      portmap    sunrpc     portmapper rusersd    100002      rusers nfs        100003      nfsprog mountd     100005      mount     showmount walld      100008      rwall     shutdown sprayd     100012      spray llockmgr   100020 nlockmgr   100021 status     100024 bootparam  100026 keyserv    100029      keyserver</pre>
<b>FILES</b>	<code>/etc/nsswitch.conf</code>
<b>SEE ALSO</b>	<code>nsswitch.conf(4)</code>

<b>NAME</b>	rpld.conf – Remote Program Load (RPL) server configuration file
<b>SYNOPSIS</b>	<code>/etc/rpld.conf</code>
<b>DESCRIPTION</b>	<p>The <code>/etc/rpld.conf</code> file contains the configuration information for operation of <b>rpld</b>, the RPL-based network boot server. It is a text file containing keyword-value pairs and comments. The keyword-value pairs specify the value to use for parameters used by the RPL server. Comments can be entered by starting the line using the # character. The user can add comments to the file for customized configurations. Alternate RPL server configuration files can be specified when running the RPL server by supplying a configuration file similar to the default configuration file.</p>
<b>Keywords</b>	<p>All keywords are case-sensitive. Not all keywords must be present. (However, note that the <b>end</b> keyword at the end of the file must be present.) If a keyword is not present, internal defaults, which are the default values described here, will be used. Keyword-value pairs are specified by:</p> <p style="padding-left: 40px;">keyword = value</p> <p><b>DebugLevel</b> Specify the number of error, warning, and information messages to be generated while the RPL server is running. The valid range is 0-9. A value of 0 means no message at all, while a value of 9 will generate the most messages. The default is 0. Note that it is best to limit the value to 8 or below; use of level 9 may generate so many debug messages that the performance of the RPL server may be impacted.</p> <p><b>DebugDest</b> A numeric value specifying where to send the messages to:</p> <p style="padding-left: 40px;">0 = standard output 1 = syslogd 2 = log file</p> <p>The default is 2.</p> <p><b>MaxClients</b> A numeric value specifying the maximum number of simultaneous network boot clients to be in service. A value of -1 means unlimited except where system resources is the limiting factor. Any positive value will set a limit on the number of clients to be in service at the same time unless system resource constraints come in before the limit. The default is -1.</p> <p><b>BackGround</b> A numeric value indicating whether the RPL server should run in the background or not. A 0 means run in the background and a 1 means do not run in the background. The difference is whether the server will relinquish the controlling terminal or not. The default is 1.</p>

**FrameSize**

The default size of data frames to be used to send bootfile data to the network boot clients. This size should not exceed the limits imposed by the underlying physical media. For **ethernet/802.3**, the maximum physical frame size is 1500 octets. The default is 1500. Note that the protocol overhead of LLC1 and RPL is 32 octets, resulting in a maximum data length of 1468 octets.

**LogFile**

The log file to which messages will be sent if **DebugDest** is set to 2 (the default). The default file is **var/spool/rpld.log**.

**StartDelay**

The initial delay factor to use to control the speed of downloading. In the default mode of operation, the downloading process does not wait for a positive acknowledgment from the client before the next data frame is sent. In the case of a fast server and slow client, data overrun can result and requests for retransmission will be frequent. By using a delay factor, the speed of data transfer is controlled to avoid retransmission requests. Note that the unit of delay is machine dependent and bears no correlation with the actual time delayed.

**DelayGran**

Delay granularity. If the initial delay factor is not suitable and the rate of downloading is either too fast or too slow, retransmission requests from the clients will be used to adjust the delay factor either upward (to slow down the data rate) or downward (to speed up the data rate). The delay granularity is used as the delay delta for adjustment.

**end**

Keyword at the end of the file. It must be present.

**FILES**

**/etc/rpld.conf**  
**/usr/sbin/rpld**

**ATTRIBUTES**

See **attributes(5)** for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Architecture	x86

**SEE ALSO**

**rpld(1M)**, **attributes(5)**

<b>NAME</b>	rt_dptbl – real-time dispatcher parameter table
<b>DESCRIPTION</b>	<p>The process scheduler (or dispatcher) is the portion of the kernel that controls allocation of the CPU to processes. The scheduler supports the notion of scheduling classes where each class defines a scheduling policy, used to schedule processes within that class. Associated with each scheduling class is a set of priority queues on which ready to run processes are linked. These priority queues are mapped by the system configuration into a set of global scheduling priorities which are available to processes within the class. (The dispatcher always selects for execution the process with the highest global scheduling priority in the system.) The priority queues associated with a given class are viewed by that class as a contiguous set of priority levels numbered from 0 (lowest priority) to <i>n</i> (highest priority—a configuration dependent value). The set of global scheduling priorities that the queues for a given class are mapped into might not start at zero and might not be contiguous (depending on the configuration).</p> <p>The real-time class maintains an in-core table, with an entry for each priority level, giving the properties of that level. This table is called the real-time dispatcher parameter table (<b>rt_dptbl</b>). The <b>rt_dptbl</b> consists of an array (<b>config_rt_dptbl[]</b>) of parameter structures (<b>struct rtdpent_t</b>), one for each of the <i>n</i> priority levels. The structure are accessed via a pointer, (<b>rt_dptbl</b>), to the array. The properties of a given priority level <i>i</i> are specified by the <i>i</i>th parameter structure in this array (<b>rt_dptbl[i]</b>).</p> <p>A parameter structure consists of the following members. These are also described in the <b>/usr/include/sys/rt.h</b> header file.</p> <p><b>rt_globpri</b> The global scheduling priority associated with this priority level. The <b>rt_globpri</b> values cannot be changed with <b>dispadmin(1M)</b>.</p> <p><b>rt_quantum</b> The length of the time quantum allocated to processes at this level in ticks (Hz). The time quantum value is only a default or starting value for processes at a particular level as the time quantum of a real-time process can be changed by the user with the <b>priocntl</b> command or the <b>priocntl</b> system call.</p> <p>An administrator can affect the behavior of the real-time portion of the scheduler by reconfiguring the <b>rt_dptbl</b>. There are two methods available for doing this: reconfigure with a loadable module at boot-time or by using <b>dispadmin(1M)</b> at run-time.</p>
<b>RT_DPTBL LOADABLE MODULE</b>	<p>The <b>rt_dptbl</b> can be reconfigured with a loadable module which contains a new real time dispatch table. The module containing the dispatch table is separate from the RT loadable module which contains the rest of the real time software. This is the only method that can be used to change the number of real time priority levels or the set of global scheduling priorities used by the real time class. The relevant procedure and source code is described in the <b>REPLACING THE RT_DPTBL LOADABLE MODULE</b> section.</p>



**DISPADMIN  
CONFIGURATION  
FILE**

The **rt\_quantum** values in the **rt\_dptbl** can be examined and modified on a running system using the **dispadmin(1M)** command. Invoking **dispadmin** for the real-time class allows the administrator to retrieve the current **rt\_dptbl** configuration from the kernel's in-core table, or overwrite the in-core table with values from a configuration file. The configuration file used for input to **dispadmin** must conform to the specific format described below.

Blank lines are ignored and any part of a line to the right of a # symbol is treated as a comment. The first non-blank, non-comment line must indicate the resolution to be used for interpreting the time quantum values. The resolution is specified as

**RES=***res*

where *res* is a positive integer between 1 and 1,000,000,000 inclusive and the resolution used is the reciprocal of *res* in seconds. (For example, **RES=1000** specifies millisecond resolution.) Although very fine (nanosecond) resolution may be specified, the time quantum lengths are rounded up to the next integral multiple of the system clock's resolution.

The remaining lines in the file are used to specify the **rt\_quantum** values for each of the real-time priority levels. The first line specifies the quantum for real-time level 0, the second line specifies the quantum for real-time level 1, etc. There must be exactly one line for each configured real-time priority level. Each **rt\_quantum** entry must be either a positive integer specifying the desired time quantum (in the resolution given by *res*), or the value -2 indicating an infinite time quantum for that level.

**EXAMPLES**

The following excerpt from a **dispadmin** configuration file illustrates the format. Note that for each line specifying a time quantum there is a comment indicating the corresponding priority level. These level numbers indicate priority within the real-time class, and the mapping between these real-time priorities and the corresponding global scheduling priorities is determined by the configuration specified in the **RT\_DPTBL** loadable module. The level numbers are strictly for the convenience of the administrator reading the file and, as with any comment, they are ignored by **dispadmin** on input. **dispadmin** assumes that the lines in the file are ordered by consecutive, increasing priority level (from 0 to the maximum configured real-time priority). The level numbers in the comments should normally agree with this ordering; if for some reason they don't, however, **dispadmin** is unaffected.

**# Real-Time Dispatcher Configuration File**

**RES=1000**

<b># TIME QUANTUM</b>	<b>PRIORITY</b>
<b># (rt_quantum)</b>	<b>LEVEL</b>
<b>100</b>	<b># 0</b>
<b>100</b>	<b># 1</b>
<b>100</b>	<b># 2</b>
<b>100</b>	<b># 3</b>
<b>100</b>	<b># 4</b>
<b>100</b>	<b># 5</b>
<b>90</b>	<b># 6</b>

```

          90          # 7
          .          . .
          .          . .
          .          . .
          10          # 58
          10          # 59

```

## REPLACING THE RT\_DPTBL LOADABLE MODULE

In order to change the size of the real time dispatch table, the loadable module which contains the dispatch table information will have to be built. It is recommended that you save the existing module before using the following procedure.

1. Place the dispatch table code shown below in a file called **rt\_dptbl.c**. An example of an **rt\_dptbl.c** file follows.
2. Compile the code using the given compilation and link lines supplied.
 

```
cc -c -O -D_KERNEL rt_dptbl.c
ld -r -o RT_DPTBL rt_dptbl.o
```
3. Copy the current dispatch table in **/usr/kernel/sched** to **RT\_DPTBL.bak**.
4. Replace the current **RT\_DPTBL** in **/usr/kernel/sched**.
5. You will have to make changes in the **/etc/system** file to reflect the changes to the sizes of the tables. See **system(4)**. The **rt\_maxpri** variable may need changing. The syntax for setting this is:

```
set RT:rt_maxpri=(class-specific value for maximum real-time priority)
```

6. Reboot the system to use the new dispatch table.

**NOTE: Great care should be used in replacing the dispatch table using this method. If you don't get it right, the system may not behave properly.**

The following is an example of a **rt\_dptbl.c** file used for building the new **rt\_dptbl**.

```

/* BEGIN rt_dptbl.c */

#include <sys/proc.h>
#include <sys/priocntl.h>
#include <sys/class.h>
#include <sys/disp.h>
#include <sys/rt.h>
#include <sys/rtpriocntl.h>

/*
 * This is the loadable module wrapper.
 */
#include <sys/modctl.h>

extern struct mod_ops mod_miscops;

```

```

/*
 * Module linkage information for the kernel.
 */

static struct modlmisc modlmisc = {
    &mod_miscops, "realtime dispatch table"
};

static struct modlinkage modlinkage = {
    MODREV_1, &modlmisc, 0
};

__init()
{
    return (mod_install(&modlinkage));
}

__info (struct modinfo *modinfop)
{
    return (mod_info(&modlinkage, modinfop));
}

rtdpent_t    config_rt_dptbl[] = {
/*  prilevel    Time quantum */
    100,        100,
    101,        100,
    102,        100,
    103,        100,
    104,        100,
    105,        100,
    106,        100,
    107,        100,
    108,        100,
    109,        100,
    110,        80,
    111,        80,
    112,        80,
    113,        80,
    114,        80,
    115,        80,
    116,        80,
    117,        80,
    118,        80,
    119,        80,

```

```
120,        60,  
121,        60,  
122,        60,  
123,        60,  
124,        60,  
125,        60,  
126,        60,  
127,        60,  
128,        60,  
129,        60,  
130,        40,  
131,        40,  
132,        40,  
133,        40,  
134,        40,  
135,        40,  
136,        40,  
137,        40,  
138,        40,  
139,        40,  
140,        20,  
141,        20,  
142,        20,  
143,        20,  
144,        20,  
145,        20,  
146,        20,  
147,        20,  
148,        20,  
149,        20,  
150,        10,  
151,        10,  
152,        10,  
153,        10,  
154,        10,  
155,        10,  
156,        10,  
157,        10,  
158,        10,  
159,        10,  
};  
  
/*  
* Return the address of config_rt_dptbl  
*/
```

```
rt_dpt_t *  
rt_getdptbl()  
{  
    return (config_rt_dptbl);  
}
```

**FILES** <sys/rt.h>

**SEE ALSO** **priocntl(1)**, **dispadm(1M)**, **priocntl(2)**, **system(4)**

*System Administration Guide*  
*System Interface Guide*

<b>NAME</b>	sbus – configuration files for SBus device drivers
<b>DESCRIPTION</b>	<p>The SBus is a geographically addressed peripheral bus present on many SPARC hardware platforms. SBus devices are <i>self-identifying</i> — that is to say the SBus card itself provides information to the system so that it can identify the device driver that needs to be used. The device usually provides additional information to the system in the form of name-value pairs that can be retrieved using the DDI property interfaces. See <b>ddi_prop_op(9F)</b> for details.</p> <p>The information is usually derived from a small Forth program stored in the FCode PROM on the card, so driver configuration files should be completely unnecessary for these devices. However, on some occasions, drivers for SBus devices may need to use driver configuration files to augment the information provided by the SBus card. See <b>driver.conf(4)</b> for further details.</p> <p>When they are needed, configuration files for SBus device drivers should identify the parent bus driver implicitly using the <i>class</i> keyword. This removes the dependency on the particular bus driver involved since this may be named differently on different platforms.</p> <p>All bus drivers of class <b>sbus</b> recognise the following properties:</p> <p><b>reg</b>                   An arbitrary length array where each element of the array consists of a 3-tuple of integers. Each array element describes a logically contiguous mappable resource on the SBus.</p> <p>                          The first integer of each tuple specifies the slot number the card is plugged into. The second integer of each 3-tuple specifies the offset in the slot address space identified by the first element. The third integer of each 3-tuple specifies the size in bytes of the mappable resource.</p> <p>                          The driver can refer to the elements of this array by index, and construct kernel mappings to these addresses using <b>ddi_map_regs(9F)</b>. The index into the array is passed as the <i>rnumber</i> argument of <b>ddi_map_regs()</b>.</p> <p><b>interrupts</b>           An arbitrary length array where each element of the array consists of a single integer. Each array element describes a possible SBus interrupt level that the device might generate.</p> <p>                          The driver can refer to the elements of this array by index, and register interrupt handlers with the system using <b>ddi_add_intr(9F)</b>. The index into the array is passed as the <i>inumber</i> argument of <b>ddi_add_intr()</b>.</p> <p><b>registers</b>           An arbitrary length array where each element of the array consists of a 3-tuple of integers. Each array element describes a logically contiguous mappable resource on the SBus.</p> <p>                          The first integer of each tuple should be set to <b>-1</b>, specifying that any SBus slot may be matched. The second integer of each 3-tuple specifies the offset in the slot address space identified by the first element. The third integer of each 3-tuple specifies the size in bytes of the mappable resource.</p>

The **registers** property can only be used to augment an incompletely specified **reg** property with information from a driver configuration file. It may only be specified in a driver configuration file.

All SBus devices must provide **reg** properties to the system. The first two integer elements of the **reg** property are used to construct the address part of the device name under **/devices**.

Only devices that generate interrupts need to provide **interrupts** properties.

Occasionally, it may be necessary to override or augment the configuration information supplied by the SBus device. This can be achieved by writing a driver configuration file that describes a prototype device information (devinfo) node specification, containing the additional properties required.

For the system to merge the information, certain conditions must be met. First, the **name** property must be the same. Second, either the first two integers (slot number and offset) of the two **reg** properties must be the same, or the second integer (offset) of the **reg** and **registers** properties must be the same.

In the event that the SBus card has no **reg** property at all, the self-identifying information cannot be used, so all the details of the card must be specified in a driver configuration file.

## EXAMPLES

Here is a configuration file for an SBus card called **SUNW,netboard**. The card already has a simple FCode PROM that creates **name** and **reg** properties, and will have a complete set of properties for normal use once the driver and firmware is complete.

In this example, we want to augment the properties given to us by the firmware. We use the same **name** property, and use the **registers** property to match the firmware **reg** property. That way we don't have to worry about which slot the card is really plugged into.

We want to add an **interrupts** property while we are developing the firmware and driver so that we can start to experiment with interrupts. The device can generate interrupts at SBus level 3. Additionally, we want to set a **debug-level** property to 4.

```
#
# Copyright (c) 1992, by Sun Microsystems, Inc.
#ident "@(#)SUNW,netboard.conf 1.4 92/03/10 SMI"
#
```

```
name="SUNW,netboard" class="sbus"
registers=-1,0x40000,64,-1,0x80000,1024
interrupts=3 debug-level=4;
```

## ATTRIBUTES

See **attributes(5)** for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Architecture	SPARC

**SEE ALSO** [driver.conf\(4\)](#), [attributes\(5\)](#), [ddi\\_add\\_intr\(9F\)](#), [ddi\\_map\\_regs\(9F\)](#), [ddi\\_prop\\_op\(9F\)](#)

*Writing Device Drivers*

**WARNINGS** The wildcarding mechanism of the **registers** property matches every instance of the particular device attached to the system. This may not always be what is wanted.



<b>NAME</b>	sccsfile – format of an SCCS history file
<b>DESCRIPTION</b>	<p>An SCCS file is an ASCII file consisting of six logical parts:</p> <ul style="list-style-type: none"> <li><i>checksum</i> character count used for error detection</li> <li><i>delta table</i> log containing version info and statistics about each delta</li> <li><i>usernames</i> login names and/or group IDs of users who may add deltas</li> <li><i>flags</i> definitions of internal keywords</li> <li><i>comments</i> arbitrary descriptive information about the file</li> <li><i>body</i> the actual text lines intermixed with control lines</li> </ul> <p>Each section is described in detail below.</p>
<b>Conventions</b>	<p>Throughout an SCCS file there are lines which begin with the ASCII SOH (start of heading) character (octal 001). This character is hereafter referred to as the <i>control character</i>, and will be represented as '^A'. If a line described below is not depicted as beginning with the control character, it cannot do so and still be within SCCS file format.</p> <p>Entries of the form <i>dddd</i> represent a five digit string (a number between 00000 and 99999).</p>
<b>Checksum</b>	<p>The checksum is the first line of an SCCS file. The form of the line is:</p> <p style="padding-left: 40px;">^A <i>hdddd</i></p> <p>The value of the checksum is the sum of all characters, except those contained in the first line. The ^Ah provides a <i>magic number</i> of (octal) 064001.</p>
<b>Delta Table</b>	<p>The delta table consists of a variable number of entries of the form:</p> <p style="padding-left: 40px;">^A<i>s inserted/deleted/unchanged</i>  ^A<i>d type sid yr/mo/da hr:mi:se username serial-number predecessor-sn</i>  ^A<i>i include-list</i>  ^A<i>x exclude-list</i>  ^A<i>g ignored-list</i>  ^A<i>m mr-number</i>  ...  ^A<i>c comments...</i>  ...  ^A<i>e</i></p> <p>The first line (^As) contains the number of lines inserted/deleted/unchanged respectively. The second line (^Ad) contains the type of the delta (normal: <b>D</b>, and removed: <b>R</b>), the SCCS ID of the delta, the date and time of creation of the delta, the user-name corresponding to the real user ID at the time the delta was created, and the serial numbers of the delta and its predecessor, respectively.</p>

	<p>The <b>^Ai</b>, <b>^Ax</b>, and <b>^Ag</b> lines contain the serial numbers of deltas included, excluded, and ignored, respectively. These lines do not always appear.</p> <p>The <b>^Am</b> lines (optional) each contain one MR number associated with the delta; the <b>^Ac</b> lines contain comments associated with the delta.</p> <p>The <b>^Ae</b> line ends the delta table entry.</p>
<b>User Names</b>	<p>The list of user-names and/or numerical group IDs of users who may add deltas to the file, separated by NEWLINE characters. The lines containing these login names and/or numerical group IDs are surrounded by the bracketing lines <b>^Au</b> and <b>^AU</b>. An empty list allows anyone to make a delta.</p>
<b>Flags</b>	<p>Flags are keywords that are used internally (see <b>sccs-admin(1)</b> for more information on their use). Each flag line takes the form:</p> <p style="padding-left: 40px;"><b>^Af</b> <i>flag optional text</i></p> <p>The following flags are defined in order of appearance:</p> <p><b>^Af t</b> <i>type-of-program</i>          Defines the replacement for the <b>%T%</b> ID keyword.</p> <p><b>^Af v</b> <i>program-name</i>          Controls prompting for MR numbers in addition to comments; if the optional text is present it defines an MR number validity checking program.</p> <p><b>^Af i</b> Indicates that the <b>'No id keywords'</b> message is to generate an error that terminates the SCCS command. Otherwise, the message is treated as a warning only.</p> <p><b>^Af b</b> Indicates that the <b>-b</b> option may be used with the SCCS <b>get</b> command to create a branch in the delta tree.</p> <p><b>^Af m</b> <i>module name</i>          Defines the first choice for the replacement text of the <b>%M%</b> ID keyword.</p> <p><b>^Af f</b> <i>floor</i>          Defines the "floor" release; the release below which no deltas may be added.</p> <p><b>^Af c</b> <i>ceiling</i>          Defines the "ceiling" release; the release above which no deltas may be added.</p> <p><b>^Af d</b> <i>default-sid</i>          The <b>d</b> flag defines the default SID to be used when none is specified on an SCCS <b>get</b> command.</p> <p><b>^Af n</b> The <b>n</b> flag enables the SCCS <b>delta</b> command to insert a "null" delta (a delta that applies <i>no</i> changes) in those releases that are skipped when a delta is made in a <i>new</i> release (for example, when delta 5.1 is made after delta 2.7, releases 3 and 4 are skipped).</p> <p><b>^Af j</b> Enables the SCCS <b>get</b> command to allow concurrent edits of the same base SID.</p>

**^Af l** *lock-releases*

Defines a *list* of releases that are locked against editing.

**^Af q** user defined

Defines the replacement for the %Q% ID keyword.

**^Af e 0 | 1**

The **e** flag indicates whether a source file is encoded or not. A **1** indicates that the file is encoded. Source files need to be encoded when they contain control characters, or when they do not end with a NEWLINE. The **e** flag allows files that contain binary data to be checked in.

**Comments** Arbitrary text surrounded by the bracketing lines **^At** and **^AT**. The comments section typically will contain a description of the file's purpose.

**Body** The body consists of text lines and control lines. Text lines do not begin with the control character, control lines do. There are three kinds of control lines: *insert*, *delete*, and *end*, represented by:

**^AI** *dddd*

**^AD** *dddd*

**^AE** *dddd*

respectively. The digit string is the serial number corresponding to the delta for the control line.

**SEE ALSO** **sccs-admin(1)**, **sccs-cdc(1)**, **sccs-comb(1)**, **sccs-delta(1)**, **sccs-get(1)**, **sccs-help(1)**, **sccs-prs(1)**, **sccs-prt(1)**, **sccs-rmdel(1)**, **sccs-sact(1)**, **sccs-sccsdiff(1)**, **sccs-unget(1)**, **sccs-val(1)**, **sccs(1)**, **what(1)**

<b>NAME</b>	scsi – configuration files for SCSI target drivers
<b>DESCRIPTION</b>	<p>The architecture of the Solaris SCSI subsystem distinguishes two types of device drivers: SCSI target drivers, and SCSI host adapter drivers. Target drivers like <b>sd</b>(7D) and <b>st</b>(7D) on SPARC and <b>cmdk</b>(7D) on x86 manage the device on the other end of the SCSI bus. Host adapter drivers manage the SCSI bus on behalf of all the devices that share it.</p> <p>Drivers for host adapters provide a common set of interfaces for target drivers. These interfaces comprise the Sun Common SCSI Architecture (SCSA) which are documented as part of the Solaris DDI/DKI. See <b>scsi_ifgetcap</b>(9F), <b>scsi_init_pkt</b>(9F), and <b>scsi_transport</b>(9F) for further details of these, and associated routines.</p> <p>Target drivers for SCSI devices should use a driver configuration file to enable them to be recognized by the system.</p> <p>Configuration files for SCSI target drivers should identify the host adapter driver implicitly using the <i>class</i> keyword to remove any dependency on the particular host adapter involved.</p> <p>All host adapter drivers of class <b>scsi</b> recognise the following properties:</p> <p><b>target</b> Integer-valued SCSI target identifier that this driver will claim.</p> <p><b>lun</b> Integer-valued SCSI logical unit number (LUN) that this driver will claim.</p> <p>All SCSI target drivers must provide <b>target</b> and <b>lun</b> properties. These properties are used to construct the address part of the device name under <b>/devices</b>.</p> <p>The SCSI target driver configuration files shipped with Solaris have entries for LUN <b>0</b> only. For devices that support other LUNs, such as some CD changers, the system administrator may edit the driver configuration file to add entries for other LUNs.</p>
<b>EXAMPLES</b>	<p>Here is a configuration file for a SCSI target driver called <b>toaster.conf</b>.</p> <pre># # Copyright (c) 1992, by Sun Microsystems, Inc. # #ident "@(#)toaster.conf 1.2 92/05/12 SMI"  name="toaster" class="scsi" target=4 lun=0;</pre> <p>Add the following lines to <b>cmdk.conf</b> for a six-CD changer on <b>target 3</b>, with LUNs <b>0</b> to <b>5</b>.</p> <pre>name="cmdk" class="scsi" target=3 lun=1; scsi_audio="sccd_sony","sccd_std"; name="cmdk" class="scsi" target=3 lun=2; scsi_audio="sccd_sony","sccd_std"; name="cmdk" class="scsi" target=3 lun=3; scsi_audio="sccd_sony","sccd_std"; name="cmdk" class="scsi" target=3 lun=4; scsi_audio="sccd_sony","sccd_std"; name="cmdk" class="scsi" target=3 lun=5;</pre>

```
scsi_audio="sccd_sony","sccd_std";
```

It is not necessary to add the line for LUN 0, as it already exists in the file shipped with Solaris.

**SEE ALSO** [driver.conf\(4\)](#), [scsi\\_ifgetcap\(9F\)](#), [scsi\\_init\\_pkt\(9F\)](#), [scsi\\_transport\(9F\)](#)

*Writing Device Drivers*

*ANSI Small Computer System Interface-2 (SCSI-2)*

**SPARC only** [sd\(7D\)](#), [st\(7D\)](#)  
**x86 only** [cmdk\(7D\)](#)

**NOTES** You need to ensure that the **target** and **lun** values claimed by your target driver do not conflict with existing target drivers on the system. For example, on SPARC, if the target is a direct access device, the standard **sd.conf** file will usually make **sd** claim it before any other driver has a chance to probe it. This is also true for x86; if the target is a direct access device, the standard **cmdk.conf** file will usually make **cmdk** claim it before any other driver has a chance to probe it.

<b>NAME</b>	securenets – configuration file for NIS security
<b>SYNOPSIS</b>	<code>/var/yp/securenets</code>
<b>DESCRIPTION</b>	<p>The <code>/var/yp/securenets</code> file defines the networks or hosts which are allowed access to information by the Network Information Service (NIS).</p> <p>The format of the file is as follows:</p> <p>Lines beginning with the ‘#’ character are treated as comments.</p> <p>Otherwise, each line contains two fields separated by white space. The first field is a netmask, the second a network.</p> <p>The netmask field may be either 255.255.255.255 or the string ‘host’ indicating that the second field is a specific host to be allowed access.</p>
<b>FILES</b>	<code>/var/yp/securenets</code> Configuration file for NIS security.
<b>SEE ALSO</b>	<code>ypserv(1M)</code> , <code>ypxfrd(1M)</code>
<b>NOTES</b>	The Network Information Service (NIS) was formerly known as Sun Yellow Pages (YP). The functionality of the two remains the same; only the name has changed. The name Yellow Pages is a registered trademark in the United Kingdom of British Telecommunications plc, and may not be used without permission.

<b>NAME</b>	services – Internet services and aliases
<b>SYNOPSIS</b>	<b>/etc/inet/services</b> <b>/etc/services</b>
<b>DESCRIPTION</b>	<p>The <b>services</b> file is a local source of information regarding each service available through the Internet. The services file can be used in conjunction with or instead of other services sources, including the NIS maps “services.byname” and the NIS+ table “services.” Programs use the <b>getservbyname(3N)</b> routines to access this information.</p> <p>The services file contains an entry for each service. Each entry has the form:</p> <pre style="margin-left: 40px;"><i>service-name port/protocol aliases</i></pre> <p><i>service-name</i>            This is the official Internet service name.</p> <p><i>port/ protocol</i>            This field is composed of the port number and protocol through which the service is provided (for instance, <b>512/tcp</b>).</p> <p><i>aliases</i>                    This is a list of alternate names by which the service might be requested.</p> <p>Fields can be separated by any number of SPACE and/or TAB characters. A ‘#’ (number sign) indicates the beginning of a comment; characters up to the end of the line are not interpreted by routines which search the file.</p> <p>Service names may contain any printable character other than a field delimiter, NEWLINE, or comment character.</p>
<b>FILES</b>	<b>/etc/nsswitch.conf</b> configuration file for name-service switch
<b>SEE ALSO</b>	<b>getservbyname(3N)</b> , <b>inetd.conf(4)</b> , <b>nsswitch.conf(4)</b>
<b>NOTES</b>	<b>/etc/inet/services</b> is the official SVR4 name of the <b>services</b> file. The symbolic link <b>/etc/services</b> exists for BSD compatibility.

<b>NAME</b>	shadow – shadow password file
<b>DESCRIPTION</b>	<p><b>/etc/shadow</b> is an access-restricted ASCII system file that stores users' encrypted passwords and related information. The shadow file can be used in conjunction with other shadow sources, including the NIS maps <b>passwd.byname</b> and <b>passwd.byuid</b> and the NIS+ table <b>passwd</b>. Programs use the <b>getspnam(3C)</b> routines to access this information. The fields for each user entry are separated by colons. Each user is separated from the next by a newline. Unlike the <b>/etc/passwd</b> file, <b>/etc/shadow</b> does not have general read permission.</p> <p>Each entry in the shadow file has the form:</p> <p style="text-align: center;"><i>username:password:lastchg:min:max:warn:inactive:expire:flag</i></p> <p>The fields are defined as follows:</p> <p><i>username</i>     The user's login name (UID).</p> <p><i>password</i>     A 13-character encrypted password for the user, a <i>lock</i> string to indicate that the login is not accessible, or no string, which shows that there is no password for the login.</p> <p><i>lastchg</i>     The number of days between January 1, 1970, and the date that the password was last modified.</p> <p><i>min</i>            The minimum number of days required between password changes.</p> <p><i>max</i>            The maximum number of days the password is valid.</p> <p><i>warn</i>           The number of days before password expires that the user is warned.</p> <p><i>inactive</i>     The number of days of inactivity allowed for that user.</p> <p><i>expire</i>        An absolute date specifying when the login may no longer be used.</p> <p><i>flag</i>            Reserved for future use, set to zero. Currently not used.</p> <p>The encrypted password consists of 13 characters chosen from a 64-character alphabet (., /, 0–9, A–Z, a–z). To update this file, use the <b>passwd(1)</b>, <b>useradd(1M)</b>, <b>usermod(1M)</b>, or <b>userdel(1M)</b> commands.</p> <p>In order to make system administration manageable, <b>/etc/shadow</b> entries should appear in exactly the same order as <b>/etc/passwd</b> entries; this includes “+” and “-” entries if the <b>compat</b> source is being used (see <b>nsswitch.conf(4)</b>).</p>
<b>FILES</b>	<p><b>/etc/shadow</b>        shadow password file</p> <p><b>/etc/passwd</b>        password file</p> <p><b>/etc/nsswitch.conf</b>    name-service switch configuration file</p>
<b>SEE ALSO</b>	<p><b>login(1)</b>, <b>passwd(1)</b>, <b>useradd(1M)</b>, <b>userdel(1M)</b>, <b>usermod(1M)</b>, <b>getspnam(3C)</b>, <b>putspent(3C)</b>, <b>nsswitch.conf(4)</b>, <b>passwd(4)</b></p>



**NOTES**

If password aging is turned on in any name service the *passwd:* line in the `/etc/nsswitch.conf` file must have a format specified in the `nsswitch.conf(4)` man page.

If the `/etc/nsswitch.conf` `passwd` policy is not in one of the supported formats, logins will not be allowed upon password expiration because the software does not know how to handle password updates under these conditions. See `nsswitch.conf(4)` for additional information.

<b>NAME</b>	sharetab – shared file system table										
<b>DESCRIPTION</b>	<p><b>sharetab</b> resides in directory <b>/etc/dfs</b> and contains a table of local resources shared by the <b>share</b> command.</p> <p>Each line of the file consists of the following fields:</p> <p style="padding-left: 2em;"><i>pathname resource fstype specific_options description</i></p> <p>where</p> <table> <tr> <td style="padding-left: 2em;"><i>pathname</i></td> <td>Indicate the path name of the shared resource.</td> </tr> <tr> <td style="padding-left: 2em;"><i>resource</i></td> <td>Indicate the symbolic name by which remote systems can access the resource.</td> </tr> <tr> <td style="padding-left: 2em;"><i>fstype</i></td> <td>Indicate the file system type of the shared resource.</td> </tr> <tr> <td style="padding-left: 2em;"><i>specific_options</i></td> <td>Indicate file-system-type-specific options that were given to the <b>share</b> command when the resource was shared.</td> </tr> <tr> <td style="padding-left: 2em;"><i>description</i></td> <td>Describe the shared resource provided by the system administrator when the resource was shared.</td> </tr> </table>	<i>pathname</i>	Indicate the path name of the shared resource.	<i>resource</i>	Indicate the symbolic name by which remote systems can access the resource.	<i>fstype</i>	Indicate the file system type of the shared resource.	<i>specific_options</i>	Indicate file-system-type-specific options that were given to the <b>share</b> command when the resource was shared.	<i>description</i>	Describe the shared resource provided by the system administrator when the resource was shared.
<i>pathname</i>	Indicate the path name of the shared resource.										
<i>resource</i>	Indicate the symbolic name by which remote systems can access the resource.										
<i>fstype</i>	Indicate the file system type of the shared resource.										
<i>specific_options</i>	Indicate file-system-type-specific options that were given to the <b>share</b> command when the resource was shared.										
<i>description</i>	Describe the shared resource provided by the system administrator when the resource was shared.										
<b>SEE ALSO</b>	<b>share(1M)</b>										

<b>NAME</b>	shells – shell database
<b>SYNOPSIS</b>	<b>/etc/shells</b>
<b>DESCRIPTION</b>	<p>The <b>shells</b> file contains a list of the shells on the system. Applications use this file to determine whether a shell is valid (see <b>getusershell(3C)</b>). For each shell a single line should be present, consisting of the shell's path, relative to root.</p> <p>A hash mark (“#”) indicates the beginning of a comment; subsequent characters up to the end of the line are not interpreted by the routines which search the file. Blank lines are also ignored.</p>
<b>FILES</b>	<b>/etc/shells</b> lists shells on system
<b>SEE ALSO</b>	<b>ftpd(1M)</b> , <b>vipw(1B)</b> , <b>getusershell(3C)</b>

**NAME** sock2path – file that maps sockets to transport providers

**SYNOPSIS** /etc/sock2path

**DESCRIPTION** The socket mapping file, /etc/sock2path, is a system file that contains the mappings between the **socket(3N)** call parameters and the transport provider driver. Its format is described on the **soconfig(1M)** manual page.

The **init(1M)** utility uses the **soconfig** utility with the **sock2path** file during the booting sequence.

**EXAMPLES** The following is a sample **sock2path** file:

#	Family	Type	Protocol	Path
	2	2	0	/dev/tcp
	2	2	6	/dev/tcp
	2	1	0	/dev/udp
	2	1	17	/dev/udp
	1	2	0	/dev/ticotsord
	1	1	0	/dev/ticlts
	2	4	0	/dev/rawip

**SEE ALSO** **soconfig(1M)**, **socket(3N)**

*Network Interfaces Programmer's Guide*

<b>NAME</b>	space – disk space requirement file
<b>DESCRIPTION</b>	<p><b>space</b> is an ASCII file that gives information about disk space requirements for the target environment. The <b>space</b> file defines space needed beyond what is used by objects defined in the <b>prototype(4)</b> file; for example, files which will be installed with the <b>installf(1M)</b> command. The <b>space</b> file should define the maximum amount of additional space that a package will require.</p> <p>The generic format of a line in this file is:</p> <p style="text-align: center;"><i>pathname blocks inodes</i></p> <p>Definitions for the fields are as follows:</p> <p><i>pathname</i> Specify a directory name which may or may not be the mount point for a filesystem. Names that do not begin with a slash (/) indicate relocatable directories.</p> <p><i>blocks</i> Define the number of disk blocks required for installation of the files and directory entries contained in the <i>pathname</i> (using a 512-byte block size).</p> <p><i>inodes</i> Define the number of inodes required for installation of the files and directory entries contained in the <i>pathname</i>.</p>
<b>EXAMPLES</b>	<pre># extra space required by config data which is # dynamically loaded onto the system data 500 1</pre>
<b>SEE ALSO</b>	<p><b>installf(1M)</b>, <b>prototype(4)</b></p> <p><i>Application Packaging Developer's Guide</i></p>

<b>NAME</b>	sulog – su command log file	
<b>SYNOPSIS</b>	<i>/var/adm/sulog</i>	
<b>DESCRIPTION</b>	<p>The <b>sulog</b> file is a record of all attempts by users on the system to execute the <b>su(1M)</b> command. Each time <b>su(1M)</b> is executed, an entry is added to the <b>sulog</b> file.</p> <p>Each entry in the <b>sulog</b> file is a single line of the form:</p> <p style="text-align: center;"><i>SU date time result port user-newuser</i></p> <p>where</p> <p><i>date</i>            The month and date <b>su(1M)</b> was executed. <i>date</i> is displayed in the form <i>mm/dd</i> where <i>mm</i> is the month number and <i>dd</i> is the day number in the month.</p> <p><i>time</i>            The time <b>su(1M)</b> was executed. <i>time</i> is displayed in the form <i>HH/MM</i> where <i>HH</i> is the hour number (24 hour system) and <i>MM</i> is the minute number.</p> <p><i>result</i>          The result of the <b>su(1M)</b> command. A '+' sign is displayed in this field if the su attempt was successful; otherwise a '-' sign is displayed.</p> <p><i>port</i>            The name of the terminal device from which <b>su(1M)</b> was executed.</p> <p><i>user</i>            The user id of the user executing the <b>su(1M)</b> command.</p> <p><i>newuser</i>        The user id being switched to with <b>su(1M)</b>.</p>	
<b>EXAMPLES</b>	<p>Here is a sample <b>sulog</b> file:</p> <pre style="margin-left: 40px;"> SU 02/25 09:29 + console root-sys SU 02/25 09:32 + pts/3 user1-root SU 03/02 08:03 + pts/5 user1-root SU 03/03 08:19 + pts/5 user1-root SU 03/09 14:24 - pts/5 guest3-root SU 03/09 14:24 - pts/5 guest3-root SU 03/14 08:31 + pts/4 user1-root </pre>	
<b>FILES</b>	<i>/var/adm/sulog</i>	su log file
	<i>/etc/default/su</i>	contains the default location of <b>sulog</b>
<b>SEE ALSO</b>	<b>su(1M)</b>	

<b>NAME</b>	sysbus, isa, eisa, mca – device tree properties for ISA, EISA, and MCA bus device drivers
<b>DESCRIPTION</b>	<p>Solaris (Intel Platform Edition) supports the ISA, EISA, and MCA buses as the system bus. Drivers for devices on these buses use the device tree built by the booting system to retrieve the necessary system resources used by the driver. These resources include device I/O port addresses, any interrupt capabilities that the device may have, any DMA channels it may require, and any memory-mapped addresses it may occupy.</p> <p>Configuration files for ISA, EISA, and MCA device drivers are only necessary to describe properties used by a particular driver that are not part of the standard properties found in the device tree. See <b>driver.conf(4)</b> for further details of configuration file syntax.</p> <p>The ISA, EISA, and MCA nexus drivers all belong to class <b>sysbus</b>. All bus drivers of class <b>sysbus</b> recognize the following properties:</p> <p><b>interrupts</b>     An arbitrary-length array where each element of the array represents a hardware interrupt (IRQ) that is used by the device. In general, this array only has one entry unless a particular device uses more than one IRQ.</p> <p>Solaris defaults all ISA, EISA, and MCA interrupts to IPL 5. This interrupt priority may be overridden by placing an <b>interrupt-priorities</b> property in a .conf file for the driver. Each entry in the array of integers for the <b>interrupt-priorities</b> property is matched one-to-one with the elements in the <b>interrupts</b> property to specify the <b>IPL</b> value that will be used by the system for this interrupt in this driver. This is the priority that this device's interrupt handler will receive relative to the interrupt handlers of other drivers. The priority is an integer from <b>1</b> to <b>16</b>. Generally, disks are assigned a priority of <b>5</b>, while mice and printers are lower, and serial communication devices are higher, typically <b>7</b>. <b>10</b> is reserved by the system and must not be used. Priorities <b>11</b> and greater are high level priorities and are generally not recommended (see <b>ddi_intr_hilevel(9F)</b>).</p> <p>The driver can refer to the elements of this array by index using <b>ddi_add_intr(9F)</b>. The index into the array is passed as the <i>inumber</i> argument of <b>ddi_add_intr()</b>.</p> <p>Only devices that generate interrupts will have an <b>interrupts</b> property.</p> <p><b>reg</b>             An arbitrary-length array where each element of the array consists of a 3-tuple of integers. Each array element describes a contiguous memory address range associated with the device on the bus.</p> <p>The first integer of the tuple specifies the memory type, <b>0</b> specifies a memory range and <b>1</b> specifies an I/O range. The second integer specifies the base address of the memory range. The third integer of each 3-tuple specifies the size, in bytes, of the mappable region.</p> <p>The driver can refer to the elements of this array by index, and construct kernel mappings to these addresses using <b>ddi_map_regs(9F)</b>. The index into the array is passed as the <i>rnumber</i> argument of <b>ddi_map_regs()</b>.</p>

All **sysbus** devices will have **reg** properties. The first tuple of this property is used to construct the address part of the device name under **/devices**. In the case of **Plug and Play ISA** devices, the first tuple is a special tuple that does not denote a memory range, but is used by the system only to create the address part of the device name. This special tuple can be recognized by determining if the top bit of the first integer is set to a one.

The order of the tuples in the **reg** property is determined by the boot system probe code and depends on the characteristics of each particular device. However, the **reg** property will maintain the same order of entries from system boot to system boot. The recommended way to determine the **reg** property for a particular device is to use the **prtconf(1M)** command after installing the particular device. The output of the **prtconf** command can be examined to determine the **reg** property for any installed device.

**dma-channels** A list of integers that specifies the DMA channels used by this device. Only devices that use DMA channels will have a **dma-channels** property.

It is recommended that drivers for devices connected to the system bus recognize the following standard property names:

**slot** The number of the slot containing the device, if known. (Only for **EISA** and **MCA** devices).

#### ATTRIBUTES

See **attributes(5)** for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Architecture	x86

#### SEE ALSO

**prtconf(1M)**, **driver.conf(4)**, **scsi(4)**, **attributes(5)**, **ddi\_add\_intr(9F)**, **ddi\_intr\_hilevel(9F)**, **ddi\_map\_regs(9F)**, **ddi\_prop\_op(9F)**

*Writing Device Drivers*



<b>NAME</b>	sysidcfg – system identification configuration file
<b>DESCRIPTION</b>	<p>When a diskless client boots for the first time or a system installs over the network, the booting software tries to obtain configuration information about the system (such as the system's root password or name service) from a <b>sysidcfg</b> file first and then the name service databases. If the booting software cannot find the information, it prompts the user to provide the appropriate information. Like the name service databases, the <b>sysidcfg</b> file can be used to avoid all the prompts and provide a totally hands-off booting process.</p> <p>The <b>sysidcfg</b> file preconfigures information through a set of keywords, and you can specify one or more of the keywords to preconfigure as much information as you want. Also, every system that requires different configuration information must have a different <b>sysidcfg</b> file. For example, you can use the same <b>sysidcfg</b> file to preconfigure the time zone for multiple systems if you want all the systems to have the same time zone configured. However, if you want to preconfigure a different root password for each of those systems, then each system would need its own <b>sysidcfg</b> file.</p>
<b>Where To Put the sysidcfg File</b>	<p>The <b>sysidcfg</b> file can reside on a shared NFS network directory or the root directory on a UFS or PCFS diskette in the system's diskette drive. If you put the <b>sysidcfg</b> file on a shared NFS network directory, you have to use the <b>-p</b> option of the <b>add_install_client(1M)</b> command (see <b>install_scripts(1M)</b>) to specify where the system being installed can find the <b>sysidcfg</b> file. If you put the <b>sysidcfg</b> file on a diskette, you need to make sure the diskette is in the system's diskette drive when the system boots (on x86 systems, the <b>sysidcfg</b> file should reside on the Solaris Device Configuration Assistant diskette).</p> <p>Only one <b>sysidcfg</b> file can reside in a directory or diskette. If you are creating more than one <b>sysidcfg</b> file, they must reside in different directories or diskettes.</p>
<b>Keyword Syntax Rules</b>	<p>The following rules apply to the keywords in a <b>sysidcfg</b> file:</p> <ul style="list-style-type: none"> <li>• Keywords can be in any order</li> <li>• Keywords are not case sensitive</li> <li>• Keyword values can be optionally enclosed in single (') or double (") quotes</li> <li>• Only the first instance of a keyword is valid; if you specify the same keyword more than once, the first keyword specified will be used.</li> </ul>

## Keywords

Platform	Configuration Information	Keywords	Where to Find Values/Example
All	Name service, domain name, name server	<b>name_service</b> =NIS, NIS+, OTHER, NONE <b>{domain_name=domain_name name_server=hostname(ip_address)}</b>	<b>name_service=NIS {domain_name=chandy.West.Arp.COM name_server=timber(129.221.2.1)}</b>
All	Network interface, host name, IP address, netmask	<b>network_interface</b> =NONE, PRIMARY, <i>value</i> <b>{hostname=host_name ip_address=ip_address netmask=netmask}</b>	<b>network_interface=le0 {hostname=feron ip_address=129.222.2.1 netmask=255.255.0.0}</b>
All	Root password	<b>root_password</b> = <i>root_password</i>	Encrypted from /etc/shadow
All	Language in which to display the install program	<b>system_locale</b> = <i>locale</i>	/usr/lib/locale
All	Terminal type	<b>terminal</b> = <i>terminal_type</i>	/usr/share/lib/terminfo/?/*
All	Time zone	<b>timezone</b> = <i>timezone</i>	/usr/share/lib/zoneinfo/*
All	Time and date	<b>timeserver</b> = <i>localhost, hostname, ip_address</i>	If you specify <b>localhost</b> as the time server, the system's time is assumed to be correct. If you specify the <i>hostname</i> or <i>ip_address</i> (if you are not running a name service) of a system, that system's time is used to set the time.
x86	Monitor type	<b>monitor</b> = <i>monitor_type</i>	Run <b>kdmconfig -d filename</b> ; append output to <b>sysidcfg</b> file
x86	Keyboard language, keyboard layout	<b>keyboard</b> = <i>keyboard_language {layout=value}</i>	Run <b>kdmconfig -d filename</b> ; append output to <b>sysidcfg</b> file
x86	Graphics card, color depth, display resolution, screen size	<b>display</b> = <i>graphics_card {size=screen_size depth=color_depth resolution=screen_resolution}</i>	Run <b>kdmconfig -d filename</b> ; append output to <b>sysidcfg</b> file
x86	Pointing device, number of buttons, IRQ level	<b>pointer</b> = <i>pointing_device {nbuttons=number_buttons irq=value}</i>	Run <b>kdmconfig -d filename</b> ; append output to <b>sysidcfg</b> file

**EXAMPLES**

The following example is a **sysidcfg** file for a group of SPARC systems to install over the network. (The host names, IP addresses, and netmask of these systems have been preconfigured by editing the name service.) Because all the system configuration information has been preconfigured, an automated installation can be created by using a custom JumpStart profile.

```
system_locale=en_US
timezone=US/Central
timeserver=localhost
terminal=sun-cmd
name_service=NIS {domain_name=marquee.central.sun.com
                 name_server=connor(129.152.112.3)}
root_password=m4QPOWNY
system_locale=C
```

The following example is a **sysidcfg** file created for a group of x86 systems to install over the network that all have the same keyboard, graphics cards, and pointing devices. The device information (keyboard, display, and pointer) was captured from running **kdmconfig -d** (see **kdmconfig(1M)**). In this example, users would see only the prompt to select a language (*system\_locale*) for displaying the rest of the Solaris installation program.

```
keyboard=ATKBD {layout=US-English}
display=ati {size=15-inch}
pointer=MS-S
timezone=US/Central
timeserver=connor
terminal=AT386
name_service=NIS {domain_name=marquee.central.sun.com
                 name_server=connor(129.152.112.3)}
root_password=URFUni9
```

**SEE ALSO**

**install\_scripts(1M)**, **kdmconfig(1M)**, **sysidtool(1M)**

*Solaris Advanced Installation Guide*

<b>NAME</b>	syslog.conf – configuration file for syslogd system log daemon																																
<b>SYNOPSIS</b>	<b>/etc/syslog.conf</b>																																
<b>DESCRIPTION</b>	<p>The file <b>/etc/syslog.conf</b> contains information used by the system log daemon, <b>syslogd(1M)</b>, to forward a system message to appropriate log files and/or users. <b>syslogd</b> preprocesses this file through <b>m4(1)</b> to obtain the correct information for certain log files, defining <b>LOGHOST</b> if the address of "loghost" is the same as one of the addresses of the host that is running <b>syslogd</b>.</p> <p>A configuration entry is composed of two TAB-separated fields:</p> <pre style="margin-left: 40px;"><i>selector</i>      <i>action</i></pre> <p>The <i>selector</i> field contains a semicolon-separated list of priority specifications of the form:</p> <pre style="margin-left: 40px;"><i>facility.level</i> [ ; <i>facility.level</i> ]</pre> <p>where <i>facility</i> is a system facility, or comma-separated list of facilities, and <i>level</i> is an indication of the severity of the condition being logged. Recognized values for <i>facility</i> include:</p> <table border="0" style="margin-left: 40px;"> <tr> <td><b>user</b></td> <td>Messages generated by user processes. This is the default priority for messages from programs or facilities not listed in this file.</td> </tr> <tr> <td><b>kern</b></td> <td>Messages generated by the kernel.</td> </tr> <tr> <td><b>mail</b></td> <td>The mail system.</td> </tr> <tr> <td><b>daemon</b></td> <td>System daemons, such as <b>in.ftpd(1M)</b></td> </tr> <tr> <td><b>auth</b></td> <td>The authorization system: <b>login(1)</b>, <b>su(1M)</b>, <b>getty(1M)</b>, among others.</td> </tr> <tr> <td><b>lpr</b></td> <td>The line printer spooling system: <b>lpr(1B)</b>, <b>lpc(1B)</b>, among others.</td> </tr> <tr> <td><b>news</b></td> <td>Reserved for the USENET network news system.</td> </tr> <tr> <td><b>uucp</b></td> <td>Reserved for the UUCP system; it does not currently use the <b>syslog</b> mechanism.</td> </tr> <tr> <td><b>cron</b></td> <td>The <b>cron/at</b> facility; <b>crontab(1)</b>, <b>at(1)</b>, <b>cron(1M)</b>, among others.</td> </tr> <tr> <td><b>local0-7</b></td> <td>Reserved for local use.</td> </tr> <tr> <td><b>mark</b></td> <td>For timestamp messages produced internally by <b>syslogd</b>.</td> </tr> <tr> <td>*</td> <td>An asterisk indicates all facilities except for the <b>mark</b> facility.</td> </tr> </table> <p>Recognized values for <i>level</i> are (in descending order of severity):</p> <table border="0" style="margin-left: 40px;"> <tr> <td><b>emerg</b></td> <td>For panic conditions that would normally be broadcast to all users.</td> </tr> <tr> <td><b>alert</b></td> <td>For conditions that should be corrected immediately, such as a corrupted system database.</td> </tr> <tr> <td><b>crit</b></td> <td>For warnings about critical conditions, such as hard device errors.</td> </tr> <tr> <td><b>err</b></td> <td>For other errors.</td> </tr> </table>	<b>user</b>	Messages generated by user processes. This is the default priority for messages from programs or facilities not listed in this file.	<b>kern</b>	Messages generated by the kernel.	<b>mail</b>	The mail system.	<b>daemon</b>	System daemons, such as <b>in.ftpd(1M)</b>	<b>auth</b>	The authorization system: <b>login(1)</b> , <b>su(1M)</b> , <b>getty(1M)</b> , among others.	<b>lpr</b>	The line printer spooling system: <b>lpr(1B)</b> , <b>lpc(1B)</b> , among others.	<b>news</b>	Reserved for the USENET network news system.	<b>uucp</b>	Reserved for the UUCP system; it does not currently use the <b>syslog</b> mechanism.	<b>cron</b>	The <b>cron/at</b> facility; <b>crontab(1)</b> , <b>at(1)</b> , <b>cron(1M)</b> , among others.	<b>local0-7</b>	Reserved for local use.	<b>mark</b>	For timestamp messages produced internally by <b>syslogd</b> .	*	An asterisk indicates all facilities except for the <b>mark</b> facility.	<b>emerg</b>	For panic conditions that would normally be broadcast to all users.	<b>alert</b>	For conditions that should be corrected immediately, such as a corrupted system database.	<b>crit</b>	For warnings about critical conditions, such as hard device errors.	<b>err</b>	For other errors.
<b>user</b>	Messages generated by user processes. This is the default priority for messages from programs or facilities not listed in this file.																																
<b>kern</b>	Messages generated by the kernel.																																
<b>mail</b>	The mail system.																																
<b>daemon</b>	System daemons, such as <b>in.ftpd(1M)</b>																																
<b>auth</b>	The authorization system: <b>login(1)</b> , <b>su(1M)</b> , <b>getty(1M)</b> , among others.																																
<b>lpr</b>	The line printer spooling system: <b>lpr(1B)</b> , <b>lpc(1B)</b> , among others.																																
<b>news</b>	Reserved for the USENET network news system.																																
<b>uucp</b>	Reserved for the UUCP system; it does not currently use the <b>syslog</b> mechanism.																																
<b>cron</b>	The <b>cron/at</b> facility; <b>crontab(1)</b> , <b>at(1)</b> , <b>cron(1M)</b> , among others.																																
<b>local0-7</b>	Reserved for local use.																																
<b>mark</b>	For timestamp messages produced internally by <b>syslogd</b> .																																
*	An asterisk indicates all facilities except for the <b>mark</b> facility.																																
<b>emerg</b>	For panic conditions that would normally be broadcast to all users.																																
<b>alert</b>	For conditions that should be corrected immediately, such as a corrupted system database.																																
<b>crit</b>	For warnings about critical conditions, such as hard device errors.																																
<b>err</b>	For other errors.																																

- warning** For warning messages.
- notice** For conditions that are not error conditions, but may require special handling. A configuration entry with a *level* value of **notice** must appear on a separate line.
- info** Informational messages.
- debug** For messages that are normally used only when debugging a program.
- none** Do not send messages from the indicated *facility* to the selected file. For example, a *selector* of

**\*.debug;mail.none**

will send all messages *except* mail messages to the selected file.

The *action* field indicates where to forward the message. Values for this field can have one of four forms:

- A filename, beginning with a leading slash, which indicates that messages specified by the *selector* are to be written to the specified file. The file will be opened in append mode.
- The name of a remote host, prefixed with an @, as with: @server, which indicates that messages specified by the *selector* are to be forwarded to the **syslogd** on the named host. The hostname "loghost" is the hostname given to the machine that will log **syslogd** messages. Every machine is "loghost" by default. See **/etc/hosts**. It is also possible to specify one machine on a network to be "loghost" by making the appropriate host table entries. If the local machine is designated to be "loghost", then **syslogd** messages are written to the appropriate files. Otherwise, they are sent to the machine "loghost" on the network.
- A comma-separated list of usernames, which indicates that messages specified by the *selector* are to be written to the named users if they are logged in.
- An asterisk, which indicates that messages specified by the *selector* are to be written to all logged-in users.

Blank lines are ignored. Lines for which the first nonwhite character is a '#' are treated as comments.

## EXAMPLES

With the following configuration file:

```
*.notice           /var/log/notice
mail.info          /var/log/notice
*.crit            /var/log/critical
kern,mark.debug   /dev/console
kern.err          @server
*.emerg          *
*.alert          root,operator
*.alert;auth.warning /var/log/auth
```

**syslogd(1M)** will log all mail system messages except **debug** messages and all **notice** (or higher) messages into a file named **/var/log/notice**. It logs all critical messages into **/var/log/critical**, and all kernel messages and 20-minute marks onto the system console. Kernel messages of **err** (error) severity or higher are forwarded to the machine named **server**. Emergency messages are forwarded to all users. The users **root** and **operator** are informed of any **alert** messages. All messages from the authorization system of **warning** level or higher are logged in the file **/var/log/auth**.

<b>FILES</b>	<b>/var/log/notice</b>	log of all mail system messages (except <b>debug</b> messages) and all messages of <b>notice</b> level or higher.
	<b>/var/log/critical</b>	log of all <b>critical</b> messages
	<b>/var/log/auth</b>	log of all messages from the authorization system of <b>warning</b> level or higher

**SEE ALSO** **at(1)**, **crontab(1)**, **logger(1)**, **login(1)**, **lp(1)**, **lpc(1B)**, **lpr(1B)**, **m4(1)**, **cron(1M)**, **getty(1M)**, **in.ftpd(1M)**, **su(1M)**, **syslogd(1M)**, **syslog(3)**, **hosts(4)**

<b>NAME</b>	system – system configuration information file
<b>DESCRIPTION</b>	<p>The <b>system</b> file is used for customizing the operation of the operating system kernel. The recommended procedure is to preserve the original <b>system</b> file before modifying it.</p> <p>The <b>system</b> file contains commands which are read by the kernel during initialization and used to customize the operation of your system. These commands are useful for modifying the system's treatment of its loadable kernel modules.</p> <p>The syntax of the <b>system</b> file consists of a list of keyword/value pairs which are recognized by the system as valid commands. Comment lines must begin with an asterisk ('*') and end with a newline character. All commands are case-insensitive except where noted. A command line can be no more than <b>80</b> characters in length.</p> <p>Commands that modify the system's operation with respect to loadable kernel modules require you to specify the module type by listing the module's namespace. The following namespaces are currently supported:</p> <ul style="list-style-type: none"> <li><b>drv</b> Modules in this namespace are device drivers.</li> <li><b>exec</b> Modules in this namespace are execution format modules. The following <b>exec</b> modules are currently provided by SunSoft: <ul style="list-style-type: none"> <li>SPARC system: <b>aoutexec</b></li> <li style="padding-left: 40px;"><b>elfexec</b></li> <li style="padding-left: 40px;"><b>intpexec</b></li> <li>x86 system: <b>coffexec</b></li> <li style="padding-left: 40px;"><b>elfexec</b></li> <li style="padding-left: 40px;"><b>intpexec</b></li> </ul> </li> <li><b>fs</b> These modules are filesystems.</li> <li><b>sched</b> These modules implement a process scheduling algorithm.</li> <li><b>strmod</b> These modules are STREAMS modules.</li> <li><b>sys</b> These modules implement loadable system-call modules.</li> <li><b>misc</b> These modules do not fit into any of the above categories, so are considered "miscellaneous" modules.</li> </ul> <p>Below is a description of each of the supported commands:</p> <p><b>exclude:</b> <i>&lt;namespace&gt;/&lt;modulename&gt;</i> Do not allow the listed loadable kernel module to be loaded. <b>exclude</b> commands are cumulative; the list of modules to <b>exclude</b> is created by combining every <b>exclude</b> entry in the <b>system</b> file.</p> <p><b>include:</b> <i>&lt;namespace&gt;/&lt;modulename&gt;</i> Include the listed loadable kernel module. This is the system's default, so using <b>include</b> does not modify the system's operation. <b>include</b> commands are cumulative.</p>

**forceload:** *<namespace>/<modulename>*

Force this kernel module to be loaded during kernel initialization. The default action is to automatically load the kernel module when its services are first accessed. **forceload** commands are cumulative.

**rootdev:** *<device name>*

Set the root device to the listed value instead of using the default root device as supplied by the boot program.

**rootfs:** *<root filesystem type>*

Set the root filesystem type to the listed value.

**moddir:** *<first module path>[:; ]<second ...>[...]*

Set the search path for loadable kernel modules. This command operates very much like the **PATH** shell variable. Multiple directories to search can be listed together, delimited either by blank spaces or colons.

**set** [*<module>:*]*<symbol>* {=, |, &} [*[-]<value>*]

Set an integer or character pointer in the kernel or in the selected kernel module to a new value. This command is used to change kernel and module parameters and thus modify the operation of your system. Assignment operations are not cumulative, whereas bitwise AND and OR operations are cumulative.

Operations that are supported for modifying integer variables are: simple assignment, inclusive bitwise OR, bitwise AND, one's complement, and negation. Variables in a specific loadable module can be targeted for modification by specifying the variable name prefixed with the kernel module name and a colon (:) separator. Values can be specified as hexadecimal (0x10), Octal (046), or Decimal (5).

The only operation supported for modifying character pointers is simple assignment. Static string data such as character arrays cannot be modified using the **set** command. Use care and ensure that the variable you are modifying is in fact a character pointer. The **set** command is very powerful, and will likely cause problems if used carelessly. The entire command, including the quoted string, cannot exceed **80** characters. The following escape sequences are supported within the quoted string:

<code>\n</code>	<b>(newline)</b>
<code>\t</code>	<b>(tab)</b>
<code>\b</code>	<b>(backspace)</b>

## EXAMPLES

The following is a sample **system** file.

```
* Force the ELF exec kernel module to be loaded during kernel
* initialization. Execution type modules are in the exec namespace.
forceload: exec/elfexec
```



\* Change the root device to `/sbus@1,f8000000/esp@0,800000/sd@3,0:a`.  
 \* You can derive root device names from `/devices`.  
 \* Root device names must be the fully expanded Open Boot Prom  
 \* device name. This command is platform and configuration specific.  
 \* This example uses the first partition (a) of the SCSI disk at  
 \* SCSI target 3 on the esp host adapter in slot 0 (on board)  
 \* of the SBus of the machine.  
 \* Adapter unit-address 3,0 at sbus unit-address 0,800000.  
**rootdev: /sbus@1,f8000000/esp@0,800000/sd@3,0:a**

\* Set the filesystem type of the root to ufs. Note that  
 \* the equal sign can be used instead of the colon.  
**rootfs:ufs**

\* Set the search path for kernel modules to look first in  
 \* `/usr/phil/mod_test` for modules, then in `/kernel/modules` (the  
 \* default) if not found. Useful for testing new modules.  
 \* Note that you can delimit your module pathnames using  
 \* colons instead of spaces: `moddir:/newmodules:/kernel/modules`  
**moddir:/usr/phil/mod\_test /kernel/modules.**

\* Set the configuration option `{_POSIX_CHOWN_RESTRICTED}` :  
 \* This configuration option is enabled by default.  
**set rstchown = 1**  
 \* Disable the configuration option `{_POSIX_CHOWN_RESTRICTED}` :  
**set rstchown = 0**

\* Set the integer variable "maxusers" in the kernel to 16. This is a  
 \* useful tuning parameter.  
**set maxusers = 16**

\* Turn on debugging messages in the modules mydriver. This is useful  
 \* during driver development.  
**set mydriver:debug = 1**

\* Bitwise AND the kernel variable "moddebug" with the  
 \* one's complement of the hex value 0x880, and set  
 \* "moddebug" to this new value.  
**set moddebug & ~0x880**

\* Demonstrate the cumulative effect of the SET  
 \* bitwise AND/OR operations by further modifying "moddebug"  
 \* by ORing it with 0x40.  
**set moddebug | 0x40**

**WARNINGS**

**system** file lines must be fewer than **80** characters in length.

Use care when modifying the **system** file; it modifies the operation of the kernel. If you preserved the original **system** file, you can use the **boot -a** option and supply the path to the original file, allowing the system to boot correctly.

**NOTES**

**/etc/system** is only read once: at boot time.

<b>NAME</b>	telnetrc – file for telnet default options
<b>DESCRIPTION</b>	<p>The <b>.telnetrc</b> file contains commands that are executed when a connection is established. Each line in the file contains a host name, one or more spaces or tabs, and a <b>telnet(1)</b> command. Lines beginning with the pound sign (#) are interpreted as comments and therefore ignored. Uppercase and lowercase are not unique in this file.</p> <p>The <b>.telnetrc</b> file is retrieved from each user's HOME directory.</p>
<b>EXAMPLES</b>	<p>A <b>.telnetrc</b> file containing the following lines:</p> <pre><b>weirdhost toggle crmod</b> <b># Always export SPRINTER</b> <b>DEFAULT environ export PRINTER</b></pre> <p>Indicates that the <b>crmod</b>, which defaults to off, should be enabled when connecting to the system <i>weirdhost</i>. In addition, the value of the environment variable PRINTER should be exported to all systems.</p>
<b>FILES</b> <b>SEE ALSO</b>	<p><b>\$HOME/.telnetrc</b> <b>telnet(1)</b>, <b>in.telnetd(1M)</b>, <b>environ(5)</b></p>

<b>NAME</b>	term – format of compiled term file
<b>SYNOPSIS</b>	<code>/usr/share/lib/terminfo/?/*</code>
<b>DESCRIPTION</b>	<p>The <b>term</b> file is compiled from <b>terminfo</b>(4) source files using <b>tic</b>(1M). Compiled files are organized in a directory hierarchy under the first letter of each terminal name. For example, the <b>vt100</b> file would have the pathname <code>/usr/lib/terminfo/v/vt100</code>. The default directory is <code>/usr/share/lib/terminfo</code>. Synonyms for the same terminal are implemented by multiple links to the same compiled file.</p> <p>The format has been chosen so that it is the same on all hardware. An 8-bit byte is assumed, but no assumptions about byte ordering or sign extension are made. Thus, these binary <b>terminfo</b> files can be transported to other hardware with 8-bit bytes.</p> <p>Short integers are stored in two 8-bit bytes. The first byte contains the least significant 8 bits of the value, and the second byte contains the most significant 8 bits. (Thus, the value represented is <math>256*second+first</math>.) The value <code>-1</code> is represented by <code>0377,0377</code>, and the value <code>-2</code> is represented by <code>0376,0377</code>; other negative values are illegal. The <code>-1</code> generally means that a capability is missing from this terminal. The <code>-2</code> means that the capability has been cancelled in the <b>terminfo</b> source and also is to be considered missing.</p> <p>The compiled file is created from the source file descriptions of the terminals (see the <code>-I</code> option of <b>infocmp</b>) by using the <b>terminfo</b> compiler, <b>tic</b>, and read by the routine <b>setup-term</b> (see <b>curses</b>(3X)). The file is divided into six parts in the following order: the header, terminal names, boolean flags, numbers, strings, and string table.</p> <p>The header section begins the file six short integers in the format described below. These integers are:</p> <ol style="list-style-type: none"> <li>1. the magic number (octal <code>0432</code>);</li> <li>2. the size, in bytes, of the names section;</li> <li>3. the number of bytes in the boolean section;</li> <li>4. the number of short integers in the numbers section;</li> <li>5. the number of offsets (short integers) in the strings section;</li> <li>6. the size, in bytes, of the string table.</li> </ol> <p>The terminal name section comes next. It contains the first line of the <b>terminfo</b> description, listing the various names for the terminal, separated by the bar ( <code> </code> ) character (see <b>term</b>(5)). The section is terminated with an ASCII NUL character.</p> <p>The terminal name section is followed by the Boolean section, number section, string section, and string table.</p> <p>The boolean flags section consists of one byte for each flag. This byte is either <code>0</code> or <code>1</code> as the flag is present or absent. The value of <code>2</code> means that the flag has been cancelled. The capabilities are in the same order as the file <code>&lt;term.h&gt;</code>.</p> <p>Between the boolean flags section and the number section, a null byte is inserted, if necessary, to ensure that the number section begins on an even byte offset. All short integers are aligned on a short word boundary.</p>

The numbers section is similar to the boolean flags section. Each capability takes up two bytes, and is stored as a short integer. If the value represented is `-1` or `-2`, the capability is taken to be missing.

The strings section is also similar. Each capability is stored as a short integer, in the format above. A value of `-1` or `-2` means the capability is missing. Otherwise, the value is taken as an offset from the beginning of the string table. Special characters in `^X` or `\c` notation are stored in their interpreted form, not the printing representation. Padding information (`$<nn>`) and parameter information (`%x`) are stored intact in uninterpreted form.

The final section is the string table. It contains all the values of string capabilities referenced in the string section. Each string is null terminated.

Note that it is possible for **setupterm** to expect a different set of capabilities than are actually present in the file. Either the database may have been updated since **setupterm** has been recompiled (resulting in extra unrecognized entries in the file) or the program may have been recompiled more recently than the database was updated (resulting in missing entries). The routine **setupterm** must be prepared for both possibilities—this is why the numbers and sizes are included. Also, new capabilities must always be added at the end of the lists of boolean, number, and string capabilities.

As an example, here is terminal information on the AT&T Model 37 KSR terminal as output by the **infocmp -I tty37** command:

```
37 | tty37 | AT&T model 37 teletype,  
    hc, os, xon,  
    bel=^G, cr=\r, cub1=\b, cud1=\n, cuu1=\E7, hd=\E9,  
    hu=\E8, ind=\n,
```

The following is an octal dump of the corresponding **term** file, produced by the **od -c /usr/share/lib/terminfo/t/tty37** command:

```

0000000 032 001      \0 032 \0 013 \0 021 001  3 \0  3  7 |  t
0000020      t  y  3  7 |  A  T  &  T      m  o  d  e  l
0000040      3  7      t  e  l  e  t  y  p  e \0 \0 \0 \0 \0
0000060  \0 \0 \0 001 \0 \0 \0 \0 \0 \0 \0 001 \0 \0 \0 \0
0000100 001 \0 \0 \0 \0 \0 377 377 377 377 377 377 377 377 377
0000120 377 377 377 377 377 377 377 377 377 377 377 377 377 & \0
0000140      \0 377 377 377 377 377 377 377 377 377 377 377 377 377
0000160 377 377  " \0 377 377 377 377  ( \0 377 377 377 377 377 377
0000200 377 377  0 \0 377 377 377 377 377 377 377 377  - \0 377 377
0000220 377 377 377 377 377 377 377 377 377 377 377 377 377 377
*
0000520 377 377 377 377 377 377 377 377 377 377 377 377 377 377 $ \0
0000540 377 377 377 377 377 377 377 377 377 377 377 377 377 377 * \0
0000560 377 377 377 377 377 377 377 377 377 377 377 377 377 377
*
0001160 377 377 377 377 377 377 377 377 377 377 377 377 377 377  3  7
0001200 |  t  t  y  3  7 |  A  T  &  T      m  o  d  e
0001220 l      3  7      t  e  l  e  t  y  p  e \0 \r \0
0001240 \n \0 \n \0 007 \0 \b \0 033  8 \0 033  9 \0 033  7
0001260 \0 \0
0001261

```

Some limitations: total compiled entries cannot exceed 4096 bytes; all entries in the name field cannot exceed 128 bytes.

**FILES** /usr/share/lib/terminfo/?/\* compiled terminal description database  
 /usr/include/term.h terminfo header  
 /usr/xpg4/include/term.h X/Open Curses terminfo header

**SEE ALSO** infocmp(1M), curses(3X), curses(3XC), terminfo(4), term(5)

<b>NAME</b>	terminfo – terminal and printer capability database
<b>SYNOPSIS</b>	<code>/usr/share/lib/terminfo/?/*</code>
<b>DESCRIPTION</b>	<p><b>terminfo</b> is a database that describes the capabilities of devices such as terminals and printers. Devices are described in <b>terminfo</b> source files by specifying a set of capabilities, by quantifying certain aspects of the device, and by specifying character sequences that effect particular results. This database is often used by screen oriented applications such as <b>vi</b> and <b>curses</b>-based programs, as well as by some system commands such as <b>ls</b> and <b>more</b>. This usage allows them to work with a variety of devices without changes to the programs.</p> <p><b>terminfo</b> descriptions are located in the directory pointed to by the environment variable <b>TERMINFO</b> or in <code>/usr/share/lib/terminfo</code>. <b>terminfo</b> descriptions are generated by <b>tic(1M)</b>.</p> <p><b>terminfo</b> source files consist of one or more device descriptions. Each description consists of a header (beginning in column 1) and one or more lines that list the features for that particular device. Every line in a <b>terminfo</b> source file must end in a comma (,). Every line in a <b>terminfo</b> source file except the header must be indented with one or more white spaces (either spaces or tabs).</p> <p>Entries in <b>terminfo</b> source files consist of a number of comma-separated fields. White space after each comma is ignored. Embedded commas must be escaped by using a backslash. Each device entry has the following format:</p> <pre style="margin-left: 40px;"> <i>alias</i><sub>1</sub>   <i>alias</i><sub>2</sub>   ...   <i>alias</i><sub>n</sub>   <i>fullname</i>,       <i>capability</i><sub>1</sub>, <i>capability</i><sub>2</sub>,       .       .       .       <i>capability</i><sub>n</sub>, </pre> <p>The first line, commonly referred to as the header line, must begin in column one and must contain at least two aliases separated by vertical bars. The last field in the header line must be the long name of the device and it may contain any string. Alias names must be unique in the <b>terminfo</b> database and they must conform to system file naming conventions (see <b>tic(1M)</b>); they cannot, for example, contain white space or slashes.</p> <p>Every device must be assigned a name, such as "vt100". Device names (except the long name) should be chosen using the following conventions. The name should not contain hyphens because hyphens are reserved for use when adding suffixes that indicate special modes.</p> <p>These special modes may be modes that the hardware can be in, or user preferences. To assign a special mode to a particular device, append a suffix consisting of a hyphen and an indicator of the mode to the device name. For example, the <b>-w</b> suffix means "wide mode"; when specified, it allows for a width of 132 columns instead of the standard 80 columns.</p>

Therefore, if you want to use a "vt100" device set to wide mode, name the device "vt100-w." Use the following suffixes where possible.

Suffix	Meaning	Example
-w	Wide mode (more than 80 columns)	5410-w
-am	With auto. margins (usually default)	vt100-am
-nam	Without automatic margins	vt100-nam
-n	Number of lines on the screen	2300-40
-na	No arrow keys (leave them in local)	c100-na
-np	Number of pages of memory	c100-4p
-rv	Reverse video	4415-rv

The **terminfo** reference manual page is organized in two sections:

- PART 1: DEVICE CAPABILITIES
- PART 2: PRINTER CAPABILITIES

**PART 1: DEVICE CAPABILITIES**

Capabilities in **terminfo** are of three types: Boolean capabilities (which show that a device has or does not have a particular feature), numeric capabilities (which quantify particular features of a device), and string capabilities (which provide sequences that can be used to perform particular operations on devices).

In the following table, a **Variable** is the name by which a C programmer accesses a capability (at the **terminfo** level). A **Capname** is the short name for a capability specified in the **terminfo** source file. It is used by a person updating the source file and by the **tput** command. A **Termcap Code** is a two-letter sequence that corresponds to the **termcap** capability name. (Note that **termcap** is no longer supported.)

Capability names have no real length limit, but an informal limit of five characters has been adopted to keep them short. Whenever possible, capability names are chosen to be the same as or similar to those specified by the ANSI X3.64-1979 standard. Semantics are also intended to match those of the ANSI standard.

All string capabilities listed below may have padding specified, with the exception of those used for input. Input capabilities, listed under the **Strings** section in the following tables, have names beginning with **key\_**. The **#i** symbol in the description field of the following tables refers to the *i*th parameter.

**Booleans**

Variable	Cap-name	Termcap Code	Description
auto_left_margin	bw	bw	<b>cub1</b> wraps from column 0 to last column
auto_right_margin	am	am	Terminal has automatic margins
back_color_erase	bce	be	Screen erased with background color
can_change	ccc	cc	Terminal can re-define existing color
ceol_standout_glitch	xhp	xs	Standout not erased by overwriting (hp)
col_addr_glitch	xhpa	YA	Only positive motion for <b>hpa</b> / <b>mhpa</b> caps
cpi_changes_res	cpix	YF	Changing character pitch changes resolution



<b>cr_cancels_micro_mode</b>	<b>crxm</b>	<b>YB</b>	Using <b>cr</b> turns off micro mode
<b>dest_tabs_magic_smso</b>	<b>xt</b>	<b>xt</b>	Destructive tabs, magic <b>smso</b> char (t1061)
<b>eat_newline_glitch</b>	<b>xenl</b>	<b>xn</b>	Newline ignored after 80 columns (Concept)
<b>erase_overstrike</b>	<b>eo</b>	<b>eo</b>	Can erase overstrikes with a blank
<b>generic_type</b>	<b>gn</b>	<b>gn</b>	Generic line type (for example, dialup, switch)
<b>hard_copy</b>	<b>hc</b>	<b>hc</b>	Hardcopy terminal
<b>hard_cursor</b>	<b>chts</b>	<b>HC</b>	Cursor is hard to see
<b>has_meta_key</b>	<b>km</b>	<b>km</b>	Has a meta key (shift, sets parity bit)
<b>has_print_wheel</b>	<b>daisy</b>	<b>YC</b>	Printer needs operator to change character set
<b>has_status_line</b>	<b>hs</b>	<b>hs</b>	Has extra "status line"
<b>hue_lightness_saturation</b>	<b>hls</b>	<b>hl</b>	Terminal uses only HLS color notation (Tektronix)
<b>insert_null_glitch</b>	<b>in</b>	<b>in</b>	Insert mode distinguishes nulls
<b>lpi_changes_res</b>	<b>lpix</b>	<b>YG</b>	Changing line pitch changes resolution
<b>memory_above</b>	<b>da</b>	<b>da</b>	Display may be retained above the screen
<b>memory_below</b>	<b>db</b>	<b>db</b>	Display may be retained below the screen
<b>move_insert_mode</b>	<b>mir</b>	<b>mi</b>	Safe to move while in insert mode
<b>move_standout_mode</b>	<b>msgr</b>	<b>ms</b>	Safe to move in standout modes
<b>needs_xon_xoff</b>	<b>nxon</b>	<b>nx</b>	Padding won't work, xon/xoff required
<b>no_esc_ctlc</b>	<b>xsb</b>	<b>xb</b>	Beehive (f1=escape, f2=ctrl C)
<b>no_pad_char</b>	<b>npc</b>	<b>NP</b>	Pad character doesn't exist
<b>non_dest_scroll_region</b>	<b>ndscr</b>	<b>ND</b>	Scrolling region is nondestructive
<b>non_rev_rmcup</b>	<b>nrrmc</b>	<b>NR</b>	<b>smcup</b> does not reverse <b>rmcup</b>
<b>over_strike</b>	<b>os</b>	<b>os</b>	Terminal overstrikes on hard-copy terminal
<b>prtr_silent</b>	<b>mc5i</b>	<b>5i</b>	Printer won't echo on screen
<b>row_addr_glitch</b>	<b>xvpa</b>	<b>YD</b>	Only positive motion for <b>vpa</b> / <b>mvpa</b> caps
<b>semi_auto_right_margin</b>	<b>sam</b>	<b>YE</b>	Printing in last column causes <b>cr</b>
<b>status_line_esc_ok</b>	<b>eslok</b>	<b>es</b>	Escape can be used on the status line
<b>tilde_glitch</b>	<b>hz</b>	<b>hz</b>	Hazeltine; can't print tilde (~)
<b>transparent_underline</b>	<b>ul</b>	<b>ul</b>	Underline character overstrikes
<b>xon_xoff</b>	<b>xon</b>	<b>xo</b>	Terminal uses xon/xoff handshaking

## Numbers

Variable	Cap-name	Termcap Code	Description
<b>bit_image_entwining</b>	<b>bitwin</b>	<b>Yo</b>	Number of passes for each bit-map row
<b>bit_image_type</b>	<b>bitype</b>	<b>Yp</b>	Type of bit image device
<b>buffer_capacity</b>	<b>bufsz</b>	<b>Ya</b>	Number of bytes buffered before printing
<b>buttons</b>	<b>btns</b>	<b>BT</b>	Number of buttons on the mouse
<b>columns</b>	<b>cols</b>	<b>co</b>	Number of columns in a line
<b>dot_horz_spacing</b>	<b>spinh</b>	<b>Yc</b>	Spacing of dots horizontally in dots per inch

<b>dot_vert_spacing</b>	<b>spinv</b>	<b>Yb</b>	Spacing of pins vertically in pins per inch
<b>init_tabs</b>	<b>it</b>	<b>it</b>	Tabs initially every # spaces
<b>label_height</b>	<b>lh</b>	<b>lh</b>	Number of rows in each label
<b>label_width</b>	<b>lw</b>	<b>lw</b>	Number of columns in each label
<b>lines</b>	<b>lines</b>	<b>li</b>	Number of lines on a screen or a page
<b>lines_of_memory</b>	<b>lm</b>	<b>lm</b>	Lines of memory if > <b>lines</b> ; <b>0</b> means varies
<b>max_attributes</b>	<b>ma</b>	<b>ma</b>	Maximum combined video attributes terminal can display
<b>magic_cookie_glitch</b>	<b>xmc</b>	<b>sg</b>	Number of blank characters left by <b>sms</b> or <b>rms</b>
<b>max_colors</b>	<b>colors</b>	<b>Co</b>	Maximum number of colors on the screen
<b>max_micro_address</b>	<b>maddr</b>	<b>Yd</b>	Maximum value in <b>micro_..._address</b>
<b>max_micro_jump</b>	<b>mjump</b>	<b>Ye</b>	Maximum value in <b>parm_..._micro</b>
<b>max_pairs</b>	<b>pairs</b>	<b>pa</b>	Maximum number of color-pairs on the screen
<b>maximum_windows</b>	<b>wnum</b>	<b>MW</b>	Maximum number of definable windows
<b>micro_char_size</b>	<b>mcs</b>	<b>Yf</b>	Character step size when in micro mode
<b>micro_line_size</b>	<b>mls</b>	<b>Yg</b>	Line step size when in micro mode
<b>no_color_video</b>	<b>ncv</b>	<b>NC</b>	Video attributes that can't be used with colors
<b>num_labels</b>	<b>nlab</b>	<b>NI</b>	Number of labels on screen (start at 1)
<b>number_of_pins</b>	<b>npins</b>	<b>Yh</b>	Number of pins in print-head
<b>output_res_char</b>	<b>orc</b>	<b>Yi</b>	Horizontal resolution in units per character
<b>output_res_line</b>	<b>orl</b>	<b>Yj</b>	Vertical resolution in units per line
<b>output_res_horz_inch</b>	<b>orhi</b>	<b>Yk</b>	Horizontal resolution in units per inch
<b>output_res_vert_inch</b>	<b>orvi</b>	<b>Yl</b>	Vertical resolution in units per inch
<b>padding_baud_rate</b>	<b>pb</b>	<b>pb</b>	Lowest baud rate where padding needed
<b>print_rate</b>	<b>cps</b>	<b>Ym</b>	Print rate in characters per second
<b>virtual_terminal</b>	<b>vt</b>	<b>vt</b>	Virtual terminal number (system)
<b>wide_char_size</b>	<b>widcs</b>	<b>Yn</b>	Character step size when in double wide mode
<b>width_status_line</b>	<b>wsl</b>	<b>ws</b>	Number of columns in status line

Strings

<b>Variable</b>	<b>Cap-name</b>	<b>Termcap Code</b>	<b>Description</b>
<b>acs_chars</b>	<b>acsc</b>	<b>ac</b>	Graphic charset pairs aAbBcC
<b>alt_scancode_esc</b>	<b>scesa</b>	<b>S8</b>	Alternate escape for scancode emulation (default is for vt100)
<b>back_tab</b>	<b>cbt</b>	<b>bt</b>	Back tab
<b>bell</b>	<b>bel</b>	<b>bl</b>	Audible signal (bell)
<b>bit_image_carriage_return</b>	<b>bicr</b>	<b>Yv</b>	Move to beginning of same row (use tparm)
<b>bit_image_newline</b>	<b>binel</b>	<b>Zz</b>	Move to next row of the bit image (use tparm)

<b>bit_image_repeat</b>	<b>birep</b>	<b>Zy</b>	Repeat bit-image cell #1 #2 times (use tparm)
<b>carriage_return</b>	<b>cr</b>	<b>cr</b>	Carriage return
<b>change_char_pitch</b>	<b>cpi</b>	<b>ZA</b>	Change number of characters per inch
<b>change_line_pitch</b>	<b>lpi</b>	<b>ZB</b>	Change number of lines per inch
<b>change_res_horz</b>	<b>chr</b>	<b>ZC</b>	Change horizontal resolution
<b>change_res_vert</b>	<b>cvr</b>	<b>ZD</b>	Change vertical resolution
<b>change_scroll_region</b>	<b>csr</b>	<b>cs</b>	Change to lines #1 through #2 (vt100)
<b>char_padding</b>	<b>rmp</b>	<b>rP</b>	Like <b>ip</b> but when in replace mode
<b>char_set_names</b>	<b>csnm</b>	<b>Zy</b>	List of character set names
<b>clear_all_tabs</b>	<b>tbc</b>	<b>ct</b>	Clear all tab stops
<b>clear_margins</b>	<b>mgc</b>	<b>MC</b>	Clear all margins (top, bottom, and sides)
<b>clear_screen</b>	<b>clear</b>	<b>cl</b>	Clear screen and home cursor
<b>clr_bol</b>	<b>el1</b>	<b>cb</b>	Clear to beginning of line, inclusive
<b>clr_eol</b>	<b>el</b>	<b>ce</b>	Clear to end of line
<b>clr_eos</b>	<b>ed</b>	<b>cd</b>	Clear to end of display
<b>code_set_init</b>	<b>csin</b>	<b>ci</b>	Init sequence for multiple codesets
<b>color_names</b>	<b>colorm</b>	<b>Yw</b>	Give name for color #1
<b>column_address</b>	<b>hpa</b>	<b>ch</b>	Horizontal position absolute
<b>command_character</b>	<b>cmdch</b>	<b>CC</b>	Terminal settable cmd character in prototype
<b>create_window</b>	<b>cwin</b>	<b>CW</b>	Define win #1 to go from #2,#3 to #4,#5
<b>cursor_address</b>	<b>cup</b>	<b>cm</b>	Move to row #1 col #2
<b>cursor_down</b>	<b>cud1</b>	<b>do</b>	Down one line
<b>cursor_home</b>	<b>home</b>	<b>ho</b>	Home cursor (if no <b>cup</b> )
<b>cursor_invisible</b>	<b>civis</b>	<b>vi</b>	Make cursor invisible
<b>cursor_left</b>	<b>cub1</b>	<b>le</b>	Move left one space.
<b>cursor_mem_address</b>	<b>mrcup</b>	<b>CM</b>	Memory relative cursor addressing
<b>cursor_normal</b>	<b>cnorm</b>	<b>ve</b>	Make cursor appear normal (undo <b>vs/vi</b> )
<b>cursor_right</b>	<b>cuf1</b>	<b>nd</b>	Non-destructive space (cursor or carriage right)
<b>cursor_to_ll</b>	<b>ll</b>	<b>ll</b>	Last line, first column (if no <b>cup</b> )
<b>cursor_up</b>	<b>cuu1</b>	<b>up</b>	Upline (cursor up)
<b>cursor_visible</b>	<b>cvvis</b>	<b>vs</b>	Make cursor very visible
<b>define_bit_image_region</b>	<b>defbi</b>	<b>Yx</b>	Define rectangular bit-image region (use tparm)
<b>define_char</b>	<b>defc</b>	<b>ZE</b>	Define a character in a character set †
<b>delete_character</b>	<b>dch1</b>	<b>dc</b>	Delete character
<b>delete_line</b>	<b>dll</b>	<b>dl</b>	Delete line
<b>device_type</b>	<b>devt</b>	<b>dv</b>	Indicate language/codeset support
<b>dial_phone</b>	<b>dial</b>	<b>DI</b>	Dial phone number #1
<b>dis_status_line</b>	<b>dsl</b>	<b>ds</b>	Disable status line

<code>display_clock</code>	<code>dclk</code>	<code>DK</code>	Display time-of-day clock
<code>display_pc_char</code>	<code>dispc</code>	<code>S1</code>	Display PC character
<code>down_half_line</code>	<code>hd</code>	<code>hd</code>	Half-line down (forward 1/2 linefeed)
<code>ena_acs</code>	<code>enacs</code>	<code>eA</code>	Enable alternate character set
<code>end_bit_image_region</code>	<code>endbi</code>	<code>Yy</code>	End a bit-image region (use tparm)
<code>enter_alt_charset_mode</code>	<code>smacs</code>	<code>as</code>	Start alternate character set
<code>enter_am_mode</code>	<code>smam</code>	<code>SA</code>	Turn on automatic margins
<code>enter_blink_mode</code>	<code>blink</code>	<code>mb</code>	Turn on blinking
<code>enter_bold_mode</code>	<code>bold</code>	<code>md</code>	Turn on bold (extra bright) mode
<code>enter_ca_mode</code>	<code>smcup</code>	<code>ti</code>	String to begin programs that use <b>cup</b>
<code>enter_delete_mode</code>	<code>smdc</code>	<code>dm</code>	Delete mode (enter)
<code>enter_dim_mode</code>	<code>dim</code>	<code>mh</code>	Turn on half-bright mode
<code>enter_doublewide_mode</code>	<code>swidm</code>	<code>ZF</code>	Enable double wide printing
<code>enter_draft_quality</code>	<code>sdrfq</code>	<code>ZG</code>	Set draft quality print mode
<code>enter_insert_mode</code>	<code>smir</code>	<code>im</code>	Insert mode (enter)
<code>enter_italics_mode</code>	<code>sitm</code>	<code>ZH</code>	Enable italics
<code>enter_leftward_mode</code>	<code>slm</code>	<code>ZI</code>	Enable leftward carriage motion
<code>enter_micro_mode</code>	<code>smicm</code>	<code>ZJ</code>	Enable micro motion capabilities
<code>enter_near_letter_quality</code>	<code>snlq</code>	<code>ZK</code>	Set near-letter quality print
<code>enter_normal_quality</code>	<code>snrmq</code>	<code>ZL</code>	Set normal quality print
<code>enter_pc_charset_mode</code>	<code>smpch</code>	<code>S2</code>	Enter PC character display mode
<code>enter_protected_mode</code>	<code>prot</code>	<code>mp</code>	Turn on protected mode
<code>enter_reverse_mode</code>	<code>rev</code>	<code>mr</code>	Turn on reverse video mode
<code>enter_scancode_mode</code>	<code>smsc</code>	<code>S4</code>	Enter PC scancode mode
<code>enter_secure_mode</code>	<code>invis</code>	<code>mk</code>	Turn on blank mode (characters invisible)
<code>enter_shadow_mode</code>	<code>sshm</code>	<code>ZM</code>	Enable shadow printing
<code>enter_standout_mode</code>	<code>smso</code>	<code>so</code>	Begin standout mode
<code>enter_subscript_mode</code>	<code>ssubm</code>	<code>ZN</code>	Enable subscript printing
<code>enter_superscript_mode</code>	<code>ssupm</code>	<code>ZO</code>	Enable superscript printing
<code>enter_underline_mode</code>	<code>smul</code>	<code>us</code>	Start underscore mode
<code>enter_upward_mode</code>	<code>sum</code>	<code>ZP</code>	Enable upward carriage motion mode
<code>enter_xon_mode</code>	<code>smxon</code>	<code>SX</code>	Turn on xon/xoff handshaking
<code>erase_chars</code>	<code>ech</code>	<code>ec</code>	Erase #1 characters
<code>exit_alt_charset_mode</code>	<code>rmacs</code>	<code>ae</code>	End alternate character set
<code>exit_am_mode</code>	<code>rmam</code>	<code>RA</code>	Turn off automatic margins
<code>exit_attribute_mode</code>	<code>sgr0</code>	<code>me</code>	Turn off all attributes
<code>exit_ca_mode</code>	<code>rmcup</code>	<code>te</code>	String to end programs that use <b>cup</b>
<code>exit_delete_mode</code>	<code>rmdc</code>	<code>ed</code>	End delete mode
<code>exit_doublewide_mode</code>	<code>rwidm</code>	<code>ZQ</code>	Disable double wide printing
<code>exit_insert_mode</code>	<code>rmir</code>	<code>ei</code>	End insert mode
<code>exit_italics_mode</code>	<code>ritm</code>	<code>ZR</code>	Disable italics
<code>exit_leftward_mode</code>	<code>rlm</code>	<code>ZS</code>	Enable rightward (normal)

<b>exit_micro_mode</b>	<b>rmicm</b>	<b>ZT</b>	carriage motion Disable micro motion capabilities
<b>exit_pc_charset_mode</b>	<b>rmpch</b>	<b>S3</b>	Disable PC character display mode
<b>exit_scancode_mode</b>	<b>rmsc</b>	<b>S5</b>	Disable PC scancode mode
<b>exit_shadow_mode</b>	<b>rshm</b>	<b>ZU</b>	Disable shadow printing
<b>exit_standout_mode</b>	<b>rmso</b>	<b>se</b>	End standout mode
<b>exit_subscript_mode</b>	<b>rsubm</b>	<b>ZV</b>	Disable subscript printing
<b>exit_superscript_mode</b>	<b>rsupm</b>	<b>ZW</b>	Disable superscript printing
<b>exit_underline_mode</b>	<b>rmul</b>	<b>ue</b>	End underscore mode
<b>exit_upward_mode</b>	<b>rum</b>	<b>ZX</b>	Enable downward (normal) carriage motion
<b>exit_xon_mode</b>	<b>rmxon</b>	<b>RX</b>	Turn off xon/xoff handshaking
<b>fixed_pause</b>	<b>pause</b>	<b>PA</b>	Pause for 2-3 seconds
<b>flash_hook</b>	<b>hook</b>	<b>fh</b>	Flash the switch hook
<b>flash_screen</b>	<b>flash</b>	<b>vb</b>	Visible bell (may not move cursor)
<b>form_feed</b>	<b>ff</b>	<b>ff</b>	Hardcopy terminal page eject
<b>from_status_line</b>	<b>fsl</b>	<b>fs</b>	Return from status line
<b>get_mouse</b>	<b>getm</b>	<b>Gm</b>	Curses should get button events
<b>goto_window</b>	<b>wingo</b>	<b>WG</b>	Go to window #1
<b>hangup</b>	<b>hup</b>	<b>HU</b>	Hang-up phone
<b>init_1string</b>	<b>is1</b>	<b>i1</b>	Terminal or printer initialization string
<b>init_2string</b>	<b>is2</b>	<b>is</b>	Terminal or printer initialization string
<b>init_3string</b>	<b>is3</b>	<b>i3</b>	Terminal or printer initialization string
<b>init_file</b>	<b>if</b>	<b>if</b>	Name of initialization file
<b>init_prog</b>	<b>ipro</b>	<b>iP</b>	Path name of program for initialization
<b>initialize_color</b>	<b>initc</b>	<b>Ic</b>	Initialize the definition of color
<b>initialize_pair</b>	<b>initp</b>	<b>Ip</b>	Initialize color-pair
<b>insert_character</b>	<b>ich1</b>	<b>ic</b>	Insert character
<b>insert_line</b>	<b>il1</b>	<b>al</b>	Add new blank line
<b>insert_padding</b>	<b>ip</b>	<b>ip</b>	Insert pad after character inserted

The “**key\_**” strings are sent by specific keys. The “**key\_**” descriptions include the macro, defined in `<curses.h>`, for the code returned by the `curses` routine `getch` when the key is pressed (see  `curs_getch(3X)`).

Variable	Cap-name	Termcap Code	Description
<b>key_a1</b>	<b>ka1</b>	<b>K1</b>	<b>KEY_A1</b> , upper left of keypad
<b>key_a3</b>	<b>ka3</b>	<b>K3</b>	<b>KEY_A3</b> , upper right of keypad
<b>key_b2</b>	<b>kb2</b>	<b>K2</b>	<b>KEY_B2</b> , center of keypad
<b>key_backspace</b>	<b>kbs</b>	<b>kb</b>	<b>KEY_BACKSPACE</b> , sent by backspace key
<b>key_beg</b>	<b>kbeg</b>	<b>@1</b>	<b>KEY_BEG</b> , sent by beg(inning) key
<b>key_btab</b>	<b>kcbt</b>	<b>kB</b>	<b>KEY_BTAB</b> , sent by back-tab key
<b>key_c1</b>	<b>kc1</b>	<b>K4</b>	<b>KEY_C1</b> , lower left of keypad

<b>key_c3</b>	<b>kc3</b>	<b>K5</b>	<b>KEY_C3</b> , lower right of keypad
<b>key_cancel</b>	<b>kcan</b>	<b>@2</b>	<b>KEY_CANCEL</b> , sent by cancel key
<b>key_catab</b>	<b>ktbc</b>	<b>ka</b>	<b>KEY_CATAB</b> , sent by clear-all-tabs key
<b>key_clear</b>	<b>kclr</b>	<b>kC</b>	<b>KEY_CLEAR</b> , sent by clear-screen or erase key
<b>key_close</b>	<b>kclo</b>	<b>@3</b>	<b>KEY_CLOSE</b> , sent by close key
<b>key_command</b>	<b>kcmd</b>	<b>@4</b>	<b>KEY_COMMAND</b> , sent by cmd (command) key
<b>key_copy</b>	<b>kcpy</b>	<b>@5</b>	<b>KEY_COPY</b> , sent by copy key
<b>key_create</b>	<b>kcrt</b>	<b>@6</b>	<b>KEY_CREATE</b> , sent by create key
<b>key_ctab</b>	<b>kctab</b>	<b>kt</b>	<b>KEY_CTAB</b> , sent by clear-tab key
<b>key_dc</b>	<b>kdch1</b>	<b>kD</b>	<b>KEY_DC</b> , sent by delete-character key
<b>key_dl</b>	<b>kdll</b>	<b>kL</b>	<b>KEY_DL</b> , sent by delete-line key
<b>key_down</b>	<b>kcud1</b>	<b>kd</b>	<b>KEY_DOWN</b> , sent by terminal down-arrow key
<b>key_eic</b>	<b>krmir</b>	<b>kM</b>	<b>KEY_EIC</b> , sent by <b>rmir</b> or <b>smir</b> in insert mode
<b>key_end</b>	<b>kend</b>	<b>@7</b>	<b>KEY_END</b> , sent by end key
<b>key_enter</b>	<b>kent</b>	<b>@8</b>	<b>KEY_ENTER</b> , sent by enter/send key
<b>key_eol</b>	<b>kel</b>	<b>kE</b>	<b>KEY_EOL</b> , sent by clear-to-end-of-line key
<b>key_eos</b>	<b>ked</b>	<b>kS</b>	<b>KEY_EOS</b> , sent by clear-to-end-of-screen key
<b>key_exit</b>	<b>kext</b>	<b>@9</b>	<b>KEY_EXIT</b> , sent by exit key
<b>key_f0</b>	<b>kf0</b>	<b>k0</b>	<b>KEY_F(0)</b> , sent by function key f0
<b>key_f1</b>	<b>kf1</b>	<b>k1</b>	<b>KEY_F(1)</b> , sent by function key f1
<b>key_f2</b>	<b>kf2</b>	<b>k2</b>	<b>KEY_F(2)</b> , sent by function key f2
<b>key_f3</b>	<b>kf3</b>	<b>k3</b>	<b>KEY_F(3)</b> , sent by function key f3
<b>key_fB</b>	<b>kf4</b>	<b>k4</b>	<b>KEY_F(4)</b> , sent by function key fB
<b>key_f5</b>	<b>kf5</b>	<b>k5</b>	<b>KEY_F(5)</b> , sent by function key f5
<b>key_f6</b>	<b>kf6</b>	<b>k6</b>	<b>KEY_F(6)</b> , sent by function key f6
<b>key_f7</b>	<b>kf7</b>	<b>k7</b>	<b>KEY_F(7)</b> , sent by function key f7
<b>key_f8</b>	<b>kf8</b>	<b>k8</b>	<b>KEY_F(8)</b> , sent by function key f8
<b>key_f9</b>	<b>kf9</b>	<b>k9</b>	<b>KEY_F(9)</b> , sent by function key f9
<b>key_f10</b>	<b>kf10</b>	<b>k;</b>	<b>KEY_F(10)</b> , sent by function key f10
<b>key_f11</b>	<b>kf11</b>	<b>F1</b>	<b>KEY_F(11)</b> , sent by function key f11
<b>key_f12</b>	<b>kf12</b>	<b>F2</b>	<b>KEY_F(12)</b> , sent by function key f12
<b>key_f13</b>	<b>kf13</b>	<b>F3</b>	<b>KEY_F(13)</b> , sent by function key f13
<b>key_f14</b>	<b>kf14</b>	<b>F4</b>	<b>KEY_F(14)</b> , sent by function key f14
<b>key_f15</b>	<b>kf15</b>	<b>F5</b>	<b>KEY_F(15)</b> , sent by function key f15
<b>key_f16</b>	<b>kf16</b>	<b>F6</b>	<b>KEY_F(16)</b> , sent by function key f16
<b>key_f17</b>	<b>kf17</b>	<b>F7</b>	<b>KEY_F(17)</b> , sent by function key f17
<b>key_f18</b>	<b>kf18</b>	<b>F8</b>	<b>KEY_F(18)</b> , sent by function key f18
<b>key_f19</b>	<b>kf19</b>	<b>F9</b>	<b>KEY_F(19)</b> , sent by function key f19
<b>key_f20</b>	<b>kf20</b>	<b>FA</b>	<b>KEY_F(20)</b> , sent by function key f20

key_f21	kf21	FB	KEY_F(21), sent by function key f21
key_f22	kf22	FC	KEY_F(22), sent by function key f22
key_f23	kf23	FD	KEY_F(23), sent by function key f23
key_f24	kf24	FE	KEY_F(24), sent by function key f24
key_f25	kf25	FF	KEY_F(25), sent by function key f25
key_f26	kf26	FG	KEY_F(26), sent by function key f26
key_f27	kf27	FH	KEY_F(27), sent by function key f27
key_f28	kf28	FI	KEY_F(28), sent by function key f28
key_f29	kf29	FJ	KEY_F(29), sent by function key f29
key_f30	kf30	FK	KEY_F(30), sent by function key f30
key_f31	kf31	FL	KEY_F(31), sent by function key f31
key_f32	kf32	FM	KEY_F(32), sent by function key f32
key_f33	kf33	FN	KEY_F(13), sent by function key f13
key_f34	kf34	FO	KEY_F(34), sent by function key f34
key_f35	kf35	FP	KEY_F(35), sent by function key f35
key_f36	kf36	FQ	KEY_F(36), sent by function key f36
key_f37	kf37	FR	KEY_F(37), sent by function key f37
key_f38	kf38	FS	KEY_F(38), sent by function key f38
key_f39	kf39	FT	KEY_F(39), sent by function key f39
key_fB0	kf40	FU	KEY_F(40), sent by function key fB0
key_fB1	kf41	FV	KEY_F(41), sent by function key fB1
key_fB2	kf42	FW	KEY_F(42), sent by function key fB2
key_fB3	kf43	FX	KEY_F(43), sent by function key fB3
key_fB4	kf44	FY	KEY_F(44), sent by function key fB4
key_fB5	kf45	FZ	KEY_F(45), sent by function key fB5
key_fB6	kf46	Fa	KEY_F(46), sent by function key fB6
key_fB7	kf47	Fb	KEY_F(47), sent by function key fB7
key_fB8	kf48	Fc	KEY_F(48), sent by function key fB8
key_fB9	kf49	Fd	KEY_F(49), sent by function key fB9
key_f50	kf50	Fe	KEY_F(50), sent by function key f50
key_f51	kf51	Ff	KEY_F(51), sent by function key f51
key_f52	kf52	Fg	KEY_F(52), sent by function key f52
key_f53	kf53	Fh	KEY_F(53), sent by function key f53
key_f54	kf54	Fi	KEY_F(54), sent by function key f54
key_f55	kf55	Fj	KEY_F(55), sent by function key f55
key_f56	kf56	Fk	KEY_F(56), sent by function key f56
key_f57	kf57	Fl	KEY_F(57), sent by function key f57
key_f58	kf58	Fm	KEY_F(58), sent by function key f58
key_f59	kf59	Fn	KEY_F(59), sent by function key f59
key_f60	kf60	Fo	KEY_F(60), sent by function key f60
key_f61	kf61	Fp	KEY_F(61), sent by function key f61
key_f62	kf62	Fq	KEY_F(62), sent by function key f62
key_f63	kf63	Fr	KEY_F(63), sent by function key f63
key_find	kfnd	@0	KEY_FIND, sent by find key
key_help	khlp	%1	KEY_HELP, sent by help key

<b>key_home</b>	<b>khome</b>	<b>kh</b>	<b>KEY_HOME</b> , sent by home key
<b>key_ic</b>	<b>kich1</b>	<b>kI</b>	<b>KEY_IC</b> , sent by ins-char/enter ins-mode key
<b>key_il</b>	<b>kil1</b>	<b>kA</b>	<b>KEY_IL</b> , sent by insert-line key
<b>key_left</b>	<b>kcub1</b>	<b>kl</b>	<b>KEY_LEFT</b> , sent by terminal left-arrow key
<b>key_ll</b>	<b>kll</b>	<b>kH</b>	<b>KEY_LL</b> , sent by home-down key
<b>key_mark</b>	<b>kmrk</b>	<b>%2</b>	<b>KEY_MARK</b> , sent by mark key
<b>key_message</b>	<b>kmsg</b>	<b>%3</b>	<b>KEY_MESSAGE</b> , sent by message key
<b>key_mouse</b>	<b>kmous</b>	<b>Km</b>	0631, Mouse event has occurred
<b>key_move</b>	<b>kmov</b>	<b>%4</b>	<b>KEY_MOVE</b> , sent by move key
<b>key_next</b>	<b>knxt</b>	<b>%5</b>	<b>KEY_NEXT</b> , sent by next-object key
<b>key_npage</b>	<b>knp</b>	<b>kN</b>	<b>KEY_NPAGE</b> , sent by next-page key
<b>key_open</b>	<b>kopn</b>	<b>%6</b>	<b>KEY_OPEN</b> , sent by open key
<b>key_options</b>	<b>kopt</b>	<b>%7</b>	<b>KEY_OPTIONS</b> , sent by options key
<b>key_ppage</b>	<b>kpp</b>	<b>kP</b>	<b>KEY_PPAGE</b> , sent by previous-page key
<b>key_previous</b>	<b>kprv</b>	<b>%8</b>	<b>KEY_PREVIOUS</b> , sent by previous-object key
<b>key_print</b>	<b>kprt</b>	<b>%9</b>	<b>KEY_PRINT</b> , sent by print or copy key
<b>key_redo</b>	<b>krdo</b>	<b>%0</b>	<b>KEY_REDO</b> , sent by redo key
<b>key_reference</b>	<b>kref</b>	<b>&amp;1</b>	<b>KEY_REFERENCE</b> , sent by reference key
<b>key_refresh</b>	<b>krfr</b>	<b>&amp;2</b>	<b>KEY_REFRESH</b> , sent by refresh key
<b>key_replace</b>	<b>krpl</b>	<b>&amp;3</b>	<b>KEY_REPLACE</b> , sent by replace key
<b>key_restart</b>	<b>krst</b>	<b>&amp;4</b>	<b>KEY_RESTART</b> , sent by restart key
<b>key_resume</b>	<b>kres</b>	<b>&amp;5</b>	<b>KEY_RESUME</b> , sent by resume key
<b>key_right</b>	<b>kcuf1</b>	<b>kr</b>	<b>KEY_RIGHT</b> , sent by terminal right-arrow key
<b>key_save</b>	<b>ksav</b>	<b>&amp;6</b>	<b>KEY_SAVE</b> , sent by save key
<b>key_sbeg</b>	<b>kBEG</b>	<b>&amp;9</b>	<b>KEY_SBEG</b> , sent by shifted beginning key
<b>key_scancel</b>	<b>kCAN</b>	<b>&amp;0</b>	<b>KEY_SCANCEL</b> , sent by shifted cancel key
<b>key_scommand</b>	<b>kCMD</b>	<b>*1</b>	<b>KEY_SCOMMAND</b> , sent by shifted command key
<b>key_scopy</b>	<b>kCPY</b>	<b>*2</b>	<b>KEY_SCOPY</b> , sent by shifted copy key
<b>key_screate</b>	<b>kCRT</b>	<b>*3</b>	<b>KEY_SCREATE</b> , sent by shifted create key
<b>key_sdc</b>	<b>kDC</b>	<b>*4</b>	<b>KEY_SDC</b> , sent by shifted delete-char key
<b>key_sdl</b>	<b>kDL</b>	<b>*5</b>	<b>KEY_SDL</b> , sent by shifted delete-line key
<b>key_select</b>	<b>kslt</b>	<b>*6</b>	<b>KEY_SELECT</b> , sent by select key
<b>key_send</b>	<b>kEND</b>	<b>*7</b>	<b>KEY_SEND</b> , sent by shifted end key
<b>key_seol</b>	<b>kEOL</b>	<b>*8</b>	<b>KEY_SEOL</b> , sent by shifted clear-line key
<b>key_sexit</b>	<b>kEXT</b>	<b>*9</b>	<b>KEY_SEXIT</b> , sent by shifted exit key
<b>key_sf</b>	<b>kind</b>	<b>kF</b>	<b>KEY_SF</b> , sent by scroll-forward/down



<b>key_sfind</b>	<b>kFND</b>	<b>*0</b>	<b>key</b> <b>KEY_SFIND</b> , sent by shifted find key
<b>key_shelp</b>	<b>kHLP</b>	<b>#1</b>	<b>KEY_SHELP</b> , sent by shifted help key
<b>key_shome</b>	<b>kHOM</b>	<b>#2</b>	<b>KEY_SHOME</b> , sent by shifted home key
<b>key_sic</b>	<b>kIC</b>	<b>#3</b>	<b>KEY_SIC</b> , sent by shifted input key
<b>key_sleft</b>	<b>kLFT</b>	<b>#4</b>	<b>KEY_SLEFT</b> , sent by shifted left-arrow key
<b>key_smessage</b>	<b>kMSG</b>	<b>%a</b>	<b>KEY_SMESSAGE</b> , sent by shifted message key
<b>key_smove</b>	<b>kMOV</b>	<b>%b</b>	<b>KEY_SMOVE</b> , sent by shifted move key
<b>key_snext</b>	<b>kNXT</b>	<b>%c</b>	<b>KEY_SNEXT</b> , sent by shifted next key
<b>key_soptions</b>	<b>kOPT</b>	<b>%d</b>	<b>KEY_SOPTIONS</b> , sent by shifted options key
<b>key_sprevious</b>	<b>kPRV</b>	<b>%e</b>	<b>KEY_SPREVIOUS</b> , sent by shifted prev key
<b>key_sprint</b>	<b>kPRT</b>	<b>%f</b>	<b>KEY_SPRINT</b> , sent by shifted print key
<b>key_sr</b>	<b>kri</b>	<b>kR</b>	<b>KEY_SR</b> , sent by scroll-backward/up key
<b>key_sredo</b>	<b>kRDO</b>	<b>%g</b>	<b>KEY_SREDO</b> , sent by shifted redo key
<b>key_sreplace</b>	<b>kRPL</b>	<b>%h</b>	<b>KEY_SREPLACE</b> , sent by shifted replace key
<b>key_sright</b>	<b>kRIT</b>	<b>%i</b>	<b>KEY_SRIGHT</b> , sent by shifted right-arrow key
<b>key_sresume</b>	<b>kRES</b>	<b>%j</b>	<b>KEY_SRSUME</b> , sent by shifted resume key
<b>key_ssave</b>	<b>kSAV</b>	<b>!1</b>	<b>KEY_SSAVE</b> , sent by shifted save key
<b>key_ssuspend</b>	<b>kSPD</b>	<b>!2</b>	<b>KEY_SSUSPEND</b> , sent by shifted suspend key
<b>key_stab</b>	<b>khts</b>	<b>kT</b>	<b>KEY_STAB</b> , sent by set-tab key
<b>key_sundo</b>	<b>kUND</b>	<b>!3</b>	<b>KEY_SUNDO</b> , sent by shifted undo key
<b>key_suspend</b>	<b>kspd</b>	<b>&amp;7</b>	<b>KEY_SUSPEND</b> , sent by suspend key
<b>key_undo</b>	<b>kund</b>	<b>&amp;8</b>	<b>KEY_UNDO</b> , sent by undo key
<b>key_up</b>	<b>kcuu1</b>	<b>ku</b>	<b>KEY_UP</b> , sent by terminal up-arrow key
<b>keypad_local</b>	<b>rmkx</b>	<b>ke</b>	Out of "keypad-transmit" mode
<b>keypad_xmit</b>	<b>smkx</b>	<b>ks</b>	Put terminal in "keypad-transmit" mode
<b>lab_f0</b>	<b>lf0</b>	<b>l0</b>	Labels on function key f0 if not f0
<b>lab_f1</b>	<b>lf1</b>	<b>l1</b>	Labels on function key f1 if not f1
<b>lab_f2</b>	<b>lf2</b>	<b>l2</b>	Labels on function key f2 if not f2
<b>lab_f3</b>	<b>lf3</b>	<b>l3</b>	Labels on function key f3 if not f3
<b>lab_fB</b>	<b>lfB</b>	<b>l4</b>	Labels on function key fB if not fB
<b>lab_f5</b>	<b>lf5</b>	<b>l5</b>	Labels on function key f5 if not f5
<b>lab_f6</b>	<b>lf6</b>	<b>l6</b>	Labels on function key f6 if not f6
<b>lab_f7</b>	<b>lf7</b>	<b>l7</b>	Labels on function key f7 if not f7
<b>lab_f8</b>	<b>lf8</b>	<b>l8</b>	Labels on function key f8 if not f8

<b>lab_f9</b>	<b>lf9</b>	<b>l9</b>	Labels on function key f9 if not f9
<b>lab_f10</b>	<b>lf10</b>	<b>l10</b>	Labels on function key f10 if not f10
<b>label_format</b>	<b>fln</b>	<b>Lf</b>	Label format
<b>label_off</b>	<b>rmln</b>	<b>LF</b>	Turn off soft labels
<b>label_on</b>	<b>smln</b>	<b>LO</b>	Turn on soft labels
<b>meta_off</b>	<b>rmm</b>	<b>mo</b>	Turn off "meta mode"
<b>meta_on</b>	<b>smm</b>	<b>mm</b>	Turn on "meta mode" (8th bit)
<b>micro_column_address</b>	<b>mhpa</b>	<b>ZY</b>	Like <b>column_address</b> for micro adjustment
<b>micro_down</b>	<b>mcud1</b>	<b>ZZ</b>	Like <b>cursor_down</b> for micro adjustment
<b>micro_left</b>	<b>mcub1</b>	<b>Za</b>	Like <b>cursor_left</b> for micro adjustment
<b>micro_right</b>	<b>mcuf1</b>	<b>Zb</b>	Like <b>cursor_right</b> for micro adjustment
<b>micro_row_address</b>	<b>mvpa</b>	<b>Zc</b>	Like <b>row_address</b> for micro adjustment
<b>micro_up</b>	<b>mcuu1</b>	<b>Zd</b>	Like <b>cursor_up</b> for micro adjustment
<b>mouse_info</b>	<b>minfo</b>	<b>Mi</b>	Mouse status information
<b>newline</b>	<b>nel</b>	<b>nw</b>	Newline (behaves like <b>cr</b> followed by <b>lf</b> )
<b>order_of_pins</b>	<b>porder</b>	<b>Ze</b>	Matches software bits to print-head pins
<b>orig_colors</b>	<b>oc</b>	<b>oc</b>	Set all color(-pair)s to the original ones
<b>orig_pair</b>	<b>op</b>	<b>op</b>	Set default color-pair to the original one
<b>pad_char</b>	<b>pad</b>	<b>pc</b>	Pad character (rather than null)
<b>parm_dch</b>	<b>dch</b>	<b>DC</b>	Delete #1 chars
<b>parm_delete_line</b>	<b>dl</b>	<b>DL</b>	Delete #1 lines
<b>parm_down_cursor</b>	<b>cud</b>	<b>DO</b>	Move down #1 lines.
<b>parm_down_micro</b>	<b>mcud</b>	<b>Zf</b>	Like <b>parm_down_cursor</b> for micro adjust.
<b>parm_ich</b>	<b>ich</b>	<b>IC</b>	Insert #1 blank chars
<b>parm_index</b>	<b>indn</b>	<b>SF</b>	Scroll forward #1 lines.
<b>parm_insert_line</b>	<b>il</b>	<b>AL</b>	Add #1 new blank lines
<b>parm_left_cursor</b>	<b>cub</b>	<b>LE</b>	Move cursor left #1 spaces
<b>parm_left_micro</b>	<b>mcub</b>	<b>Zg</b>	Like <b>parm_left_cursor</b> for micro adjust.
<b>parm_right_cursor</b>	<b>cuf</b>	<b>RI</b>	Move right #1 spaces.
<b>parm_right_micro</b>	<b>mcuf</b>	<b>Zh</b>	Like <b>parm_right_cursor</b> for micro adjust.
<b>parm_rindex</b>	<b>rin</b>	<b>SR</b>	Scroll backward #1 lines.
<b>parm_up_cursor</b>	<b>cuu</b>	<b>UP</b>	Move cursor up #1 lines.
<b>parm_up_micro</b>	<b>mcuu</b>	<b>Zi</b>	Like <b>parm_up_cursor</b> for micro adjust.
<b>pc_term_options</b>	<b>pctrm</b>	<b>S6</b>	PC terminal options
<b>pkey_key</b>	<b>pfkey</b>	<b>pk</b>	Prog funct key #1 to type string #2
<b>pkey_local</b>	<b>pfloc</b>	<b>pl</b>	Prog funct key #1 to execute string #2
<b>pkey_plab</b>	<b>pfxl</b>	<b>xl</b>	Prog key #1 to xmit string #2 and show string #3
<b>pkey_xmit</b>	<b>pfx</b>	<b>px</b>	Prog funct key #1 to xmit string #2

<b>plab_norm</b>	<b>pln</b>	<b>pn</b>	Prog label #1 to show string #2
<b>print_screen</b>	<b>mc0</b>	<b>ps</b>	Print contents of the screen
<b>prtr_non</b>	<b>mc5p</b>	<b>pO</b>	Turn on the printer for #1 bytes
<b>prtr_off</b>	<b>mc4</b>	<b>pf</b>	Turn off the printer
<b>prtr_on</b>	<b>mc5</b>	<b>po</b>	Turn on the printer
<b>pulse</b>	<b>pulse</b>	<b>PU</b>	Select pulse dialing
<b>quick_dial</b>	<b>qdial</b>	<b>QD</b>	Dial phone number #1, without progress detection
<b>remove_clock</b>	<b>rmclk</b>	<b>RC</b>	Remove time-of-day clock
<b>repeat_char</b>	<b>rep</b>	<b>rp</b>	Repeat char #1 #2 times
<b>req_for_input</b>	<b>rfi</b>	<b>RF</b>	Send next input char (for ptys)
<b>req_mouse_pos</b>	<b>reqmp</b>	<b>RQ</b>	Request mouse position report
<b>reset_1string</b>	<b>rs1</b>	<b>r1</b>	Reset terminal completely to sane modes
<b>reset_2string</b>	<b>rs2</b>	<b>r2</b>	Reset terminal completely to sane modes
<b>reset_3string</b>	<b>rs3</b>	<b>r3</b>	Reset terminal completely to sane modes
<b>reset_file</b>	<b>rf</b>	<b>rf</b>	Name of file containing reset string
<b>restore_cursor</b>	<b>rc</b>	<b>rc</b>	Restore cursor to position of last sc
<b>row_address</b>	<b>vpa</b>	<b>cv</b>	Vertical position absolute
<b>save_cursor</b>	<b>sc</b>	<b>sc</b>	Save cursor position
<b>scancode_escape</b>	<b>scesc</b>	<b>S7</b>	Escape for scancode emulation
<b>scroll_forward</b>	<b>ind</b>	<b>sf</b>	Scroll text up
<b>scroll_reverse</b>	<b>ri</b>	<b>sr</b>	Scroll text down
<b>select_char_set</b>	<b>scs</b>	<b>Zj</b>	Select character set
<b>set0_des_seq</b>	<b>s0ds</b>	<b>s0</b>	Shift into codeset 0 (EUC set 0, ASCII)
<b>set1_des_seq</b>	<b>s1ds</b>	<b>s1</b>	Shift into codeset 1
<b>set2_des_seq</b>	<b>s2ds</b>	<b>s2</b>	Shift into codeset 2
<b>set3_des_seq</b>	<b>s3ds</b>	<b>s3</b>	Shift into codeset 3 attributes #1-#6
<b>set_a_background</b>	<b>setab</b>	<b>AB</b>	Set background color using ANSI escape
<b>set_a_foreground</b>	<b>setaf</b>	<b>AF</b>	Set foreground color using ANSI escape
<b>set_attributes</b>	<b>sgr</b>	<b>sa</b>	Define the video attributes #1-#9
<b>set_background</b>	<b>setb</b>	<b>Sb</b>	Set current background color
<b>set_bottom_margin</b>	<b>smgb</b>	<b>Zk</b>	Set bottom margin at current line
<b>set_bottom_margin_parm</b>	<b>smgbp</b>	<b>Zl</b>	Set bottom margin at line #1 or #2 lines from bottom
<b>set_clock</b>	<b>sclk</b>	<b>SC</b>	Set time-of-day clock
<b>set_color_band</b>	<b>setcolor</b>	<b>Yz</b>	Change to ribbon color #1
<b>set_color_pair</b>	<b>scp</b>	<b>sp</b>	Set current color-pair
<b>set_foreground</b>	<b>setf</b>	<b>Sf</b>	Set current foreground color1
<b>set_left_margin</b>	<b>smgl</b>	<b>ML</b>	Set left margin at current line
<b>set_left_margin_parm</b>	<b>smglp</b>	<b>Zm</b>	Set left (right) margin at column #1 (#2)
<b>set_lr_margin</b>	<b>smglr</b>	<b>ML</b>	Sets both left and right margins
<b>set_page_length</b>	<b>slines</b>	<b>YZ</b>	Set page length to #1 lines (use tparm) of an inch
<b>set_right_margin</b>	<b>smgr</b>	<b>MR</b>	Set right margin at current column

<b>set_right_margin_parm</b>	<b>smgrp</b>	<b>Zn</b>	Set right margin at column #1
<b>set_tab</b>	<b>hts</b>	<b>st</b>	Set a tab in all rows, current column
<b>set_tb_margin</b>	<b>smgtb</b>	<b>MT</b>	Sets both top and bottom margins
<b>set_top_margin</b>	<b>smgt</b>	<b>Zo</b>	Set top margin at current line
<b>set_top_margin_parm</b>	<b>smgtp</b>	<b>Zp</b>	Set top (bottom) margin at line #1 (#2)
<b>set_window</b>	<b>wind</b>	<b>wi</b>	Current window is lines #1-#2 cols #3-#4
<b>start_bit_image</b>	<b>sbim</b>	<b>Zq</b>	Start printing bit image graphics
<b>start_char_set_def</b>	<b>scsd</b>	<b>Zr</b>	Start definition of a character set
<b>stop_bit_image</b>	<b>rbim</b>	<b>Zs</b>	End printing bit image graphics
<b>stop_char_set_def</b>	<b>rcsd</b>	<b>Zt</b>	End definition of a character set
<b>subscript_characters</b>	<b>subcs</b>	<b>Zu</b>	List of "subscript-able" characters
<b>superscript_characters</b>	<b>supcs</b>	<b>Zv</b>	List of "superscript-able" characters
<b>tab</b>	<b>ht</b>	<b>ta</b>	Tab to next 8-space hardware tab stop
<b>these_cause_cr</b>	<b>docr</b>	<b>Zw</b>	Printing any of these chars causes <b>cr</b>
<b>to_status_line</b>	<b>tsl</b>	<b>ts</b>	Go to status line, col #1
<b>tone</b>	<b>tone</b>	<b>TO</b>	Select touch tone dialing
<b>user0</b>	<b>u0</b>	<b>u0</b>	User string 0
<b>user1</b>	<b>u1</b>	<b>u1</b>	User string 1
<b>user2</b>	<b>u2</b>	<b>u2</b>	User string 2
<b>user3</b>	<b>u3</b>	<b>u3</b>	User string 3
<b>user4</b>	<b>u4</b>	<b>u4</b>	User string 4
<b>user5</b>	<b>u5</b>	<b>u5</b>	User string 5
<b>user6</b>	<b>u6</b>	<b>u6</b>	User string 6
<b>user7</b>	<b>u7</b>	<b>u7</b>	User string 7
<b>user8</b>	<b>u8</b>	<b>u8</b>	User string 8
<b>user9</b>	<b>u9</b>	<b>u9</b>	User string 9
<b>underline_char</b>	<b>uc</b>	<b>uc</b>	Underscore one char and move past it
<b>up_half_line</b>	<b>hu</b>	<b>hu</b>	Half-line up (reverse 1/2 linefeed)
<b>wait_tone</b>	<b>wait</b>	<b>WA</b>	Wait for dial tone
<b>xoff_character</b>	<b>xoffc</b>	<b>XF</b>	X-off character
<b>xon_character</b>	<b>xonc</b>	<b>XN</b>	X-on character
<b>zero_motion</b>	<b>zerom</b>	<b>Zx</b>	No motion for the subsequent character

**Sample Entry**

The following entry, which describes the AT&T 610 terminal, is among the more complex entries in the **terminfo** file as of this writing.

```

610 | 610bct | ATT610 | att610 | AT&T 610; 80 column; 98key keyboard
am, eslok, hs, mir, msgr, xenl, xon,
cols#80, it#8, lh#2, lines#24, lw#8, nlab#8, wsl#80,
acsc="aaffggjjkllmmnnooppqrrssttuuvvwxxyzz{| |}~",
bel=^G, blink=\E[5m, bold=\E[1m, cbt=\E[Z,
civis=\E[?25l, clear=\E[H\E[J, cnorm=\E[?25h\E[?12l,
cr=\r, csr=\E[%i%p1%d;%p2%dr, cub=\E[%p1%dD, cub1=\b,
cud=\E[%p1%dB, cud1=\E[B, cuf=\E[%p1%dC, cuf1=\E[C,
cup=\E[%i%p1%d;%p2%dH, cuu=\E[%p1%dA, cuu1=\E[A,

```

```

cvvis=\E[?12;25h, dch=\E[%p1%dP, dch1=\E[P, dim=\E[2m,
dl=\E[%p1%dM, dl1=\E[M, ed=\E[J, el=\E[K, el1=\E[1K,
flash=\E[?5h$<200>\E[?5l, fsl=\E8, home=\E[H, ht=\t,
ich=\E[%p1%d@, il=\E[%p1%dL, il1=\E[L, ind=\ED, .ind=\ED$<9>,
invis=\E[8m,
is1=\E[8;0 | \E[?3;4;5;13;15l\E[13;20l\E[?7h\E[12h\E(B\E)0,
is2=\E[0m^O, is3=\E(B\E)0, kLFT=\E[\s@, kRIT=\E[\sA,
kbs=^H, kcbt=\E[Z, kclr=\E[2J, kcub1=\E[D, kcud1=\E[B,
kcu1=\E[C, kcuu1=\E[A, kf1=\EOc, kf10=\ENp,
kf11=\ENq, kf12=\ENr, kf13=\ENs, kf14=\ENT, kf2=\EOd,
kf3=\EOe, kf4=\EOf, kf5=\EOg, kf6=\EOh, kf7=\EOi,
kf8=\EOj, kf9=\ENo, khome=\E[H, kind=\E[S, kri=\E[T,
ll=\E[24H, mc4=\E[?4i, mc5=\E[?5i, nel=\EE,
pfxl=\E[%p1%d;%p2%l%02dq%?%p1%{9}%<%t\s\s\sF%p1%1d\s\s\s\s
\s\s\s\s\s%;%p2%s,
pln=\E[%p1%d;0;0q%p2%:-16.16s, rc=\E8, rev=\E[7m,
ri=\EM, rmacs=^O, rmir=\E[4l, rmln=\E[2p, rmso=\E[m,
rmul=\E[m, rs2=\Ec\E[?3l, sc=\E7,
sgr=\E[0%?%p6%t;1%;%?%p5%t;2%;%?%p2%t;4%;%?%p4%t;5%;
%?%p3%p1% | %t;7%;%?%p7%t;8%;m%?%p9%t^N%e^O%;,
sgr0=\E[m^O, smacs=^N, smir=\E[4h, smln=\E[p,
smso=\E[7m, smul=\E[4m, tsl=\E7\E[25;%i%p1%dx,

```

### Types of Capabilities in the Sample Entry

The sample entry shows the formats for the three types of **terminfo** capabilities listed: Boolean, numeric, and string. All capabilities specified in the **terminfo** source file must be followed by commas, including the last capability in the source file. In **terminfo** source files, capabilities are referenced by their capability names (as shown in the previous tables).

Boolean capabilities are specified simply by their comma separated cap names.

Numeric capabilities are followed by the character '#' and then a positive integer value. Thus, in the sample, **cols** (which shows the number of columns available on a device) is assigned the value **80** for the AT&T 610. (Values for numeric capabilities may be specified in decimal, octal, or hexadecimal, using normal C programming language conventions.)

Finally, string-valued capabilities such as **el** (clear to end of line sequence) are listed by a two- to five-character capname, an '=', and a string ended by the next occurrence of a comma. A delay in milliseconds may appear anywhere in such a capability, preceded by \$ and enclosed in angle brackets, as in **el**=\EK\$<3>. Padding characters are supplied by **tput**. The delay can be any of the following: a number, a number followed by an asterisk, such as 5\*, a number followed by a slash, such as 5/, or a number followed by both, such as 5\*/. A '\*' shows that the padding required is proportional to the number of lines

affected by the operation, and the amount given is the per-affected-unit padding required. (In the case of insert characters, the factor is still the number of lines affected. This is always 1 unless the device has **in** and the software uses it.) When a '\*' is specified, it is sometimes useful to give a delay of the form 3.5 to specify a delay per unit to tenths of milliseconds. (Only one decimal place is allowed.)

A '/' indicates that the padding is mandatory. If a device has **xon** defined, the padding information is advisory and will only be used for cost estimates or when the device is in raw mode. Mandatory padding will be transmitted regardless of the setting of **xon**. If padding (whether advisory or mandatory) is specified for **bel** or **flash**, however, it will always be used, regardless of whether **xon** is specified.

**terminfo** offers notation for encoding special characters. Both **\E** and **\e** map to an ESCAPE character, **\x** maps to a control *x* for any appropriate *x*, and the sequences **\n**, **\l**, **\r**, **\t**, **\b**, **\f**, and **\s** give a newline, linefeed, return, tab, backspace, formfeed, and space, respectively. Other escapes include: **\^** for caret (^); **\** for backslash (\); **\,** for comma (,); **\:** for colon (:); and **\0** for null. (**\0** will actually produce **\200**, which does not terminate a string but behaves as a null character on most devices, providing CS7 is specified. (See **stty(1)**). Finally, characters may be given as three octal digits after a backslash (for example, **\123**).

Sometimes individual capabilities must be commented out. To do this, put a period before the capability name. For example, see the second **ind** in the example above. Note that capabilities are defined in a left-to-right order and, therefore, a prior definition will override a later definition.

### Preparing Descriptions

The most effective way to prepare a device description is by imitating the description of a similar device in **terminfo** and building up a description gradually, using partial descriptions with **vi** to check that they are correct. Be aware that a very unusual device may expose deficiencies in the ability of the **terminfo** file to describe it or the inability of **vi** to work with that device. To test a new device description, set the environment variable **TERMINFO** to the pathname of a directory containing the compiled description you are working on and programs will look there rather than in **/usr/share/lib/terminfo**. To get the padding for insert-line correct (if the device manufacturer did not document it) a severe test is to comment out **xon**, edit a large file at 9600 baud with **vi**, delete 16 or so lines from the middle of the screen, and then press the **u** key several times quickly. If the display is corrupted, more padding is usually needed. A similar test can be used for insert-character.

### Section 1-1: Basic Capabilities

The number of columns on each line for the device is given by the **cols** numeric capability. If the device has a screen, then the number of lines on the screen is given by the **lines** capability. If the device wraps around to the beginning of the next line when it reaches the right margin, then it should have the **am** capability. If the terminal can clear its screen, leaving the cursor in the home position, then this is given by the **clear** string

capability. If the terminal overstrikes (rather than clearing a position when a character is struck over) then it should have the **os** capability. If the device is a printing terminal, with no soft copy unit, specify both **hc** and **os**. If there is a way to move the cursor to the left edge of the current row, specify this as **cr**.

(Normally this will be carriage return, control M.) If there is a way to produce an audible signal (such as a bell or a beep), specify it as **bel**. If, like most devices, the device uses the xon-xoff flow-control protocol, specify **xon**.

If there is a way to move the cursor one position to the left (such as backspace), that capability should be given as **cub1**. Similarly, sequences to move to the right, up, and down should be given as **cuf1**, **cuu1**, and **cud1**, respectively. These local cursor motions must not alter the text they pass over; for example, you would not normally use “**cuf1**=\s” because the space would erase the character moved over.

A very important point here is that the local cursor motions encoded in **terminfo** are undefined at the left and top edges of a screen terminal. Programs should never attempt to backspace around the left edge, unless **bw** is specified, and should never attempt to go up locally off the top. To scroll text up, a program goes to the bottom left corner of the screen and sends the **ind** (index) string.

To scroll text down, a program goes to the top left corner of the screen and sends the **ri** (reverse index) string. The strings **ind** and **ri** are undefined when not on their respective corners of the screen.

Parameterized versions of the scrolling sequences are **indn** and **rin**. These versions have the same semantics as **ind** and **ri**, except that they take one parameter and scroll the number of lines specified by that parameter. They are also undefined except at the appropriate edge of the screen.

The **am** capability tells whether the cursor sticks at the right edge of the screen when text is output, but this does not necessarily apply to a **cuf1** from the last column. Backward motion from the left edge of the screen is possible only when **bw** is specified. In this case, **cub1** will move to the right edge of the previous row. If **bw** is not given, the effect is undefined. This is useful for drawing a box around the edge of the screen, for example. If the device has switch selectable automatic margins, **am** should be specified in the **terminfo** source file. In this case, initialization strings should turn on this option, if possible. If the device has a command that moves to the first column of the next line, that command can be given as **nel** (newline). It does not matter if the command clears the remainder of the current line, so if the device has no **cr** and **lf** it may still be possible to craft a working **nel** out of one or both of them.

These capabilities suffice to describe hardcopy and screen terminals. Thus the AT&T 5320 hardcopy terminal is described as follows:

```
5320 | att5320 | AT&T 5320 hardcopy terminal,
    am, hc, os,
    cols#132,
```

```

bel=^G, cr=\r, cub1=\b, cnd1=\n,
dch1=\E[P, dl1=\E[M,
ind=\n,

```

while the Lear Siegler ADM-3 is described as

```

adm3 | lsi adm3,
am, bel=^G, clear=^Z, cols#80, cr=^M, cub1=^H,
cul1=^J, ind=^J, lines#24,

```

### Section 1-2: Parameterized Strings

Cursor addressing and other strings requiring parameters are described by a parameterized string capability, with **printf**-like escapes (**%x**) in it. For example, to address the cursor, the **cup** capability is given, using two parameters: the row and column to address to. (Rows and columns are numbered from zero and refer to the physical screen visible to the user, not to any unseen memory.) If the terminal has memory relative cursor addressing, that can be indicated by **mrkup**.

The parameter mechanism uses a stack and special **%** codes to manipulate the stack in the manner of Reverse Polish Notation (postfix). Typically a sequence will push one of the parameters onto the stack and then print it in some format. Often more complex operations are necessary. Operations are in postfix form with the operands in the usual order. That is, to subtract 5 from the first parameter, one would use **%p1%{5}%-**.

The **%** encodings have the following meanings:

```

%%      outputs '%'
%[:]flags[width.precision][doxXs]
        as in printf, flags are [-+#] and space
%c      print pop gives %c
%p[1-9]
        push ith parm
%P[a-z]
        set dynamic variable [a-z] to pop
%g[a-z]
        get dynamic variable [a-z] and push it
%P[A-Z]
        set static variable [a-z] to pop
%g[A-Z]
        get static variable [a-z] and push it
%'c'   push char constant c
%{nn}  push decimal constant nn
%l      push strlen(pop)
%+ %- %* %/ %m

```



arithmetic (**%m** is mod): push(pop \$integer sub 2\$ op pop \$integer sub 1\$)

**%& %| %^**  
bit operations: push(pop \$integer sub 2\$ op pop \$integer sub 1\$)

**%= %> %<**  
logical operations: push(pop \$integer sub 2\$ op pop \$integer sub 1\$)

**%A %O**  
logical operations: and, or

**%! %~** unary operations: push(op pop)

**%i** (for ANSI terminals) add 1 to first parm, if one parm present, or first two parms, if more than one parm present

**?? expr %t thenpart %e elsepart %;**  
if-then-else, **%e elsepart** is optional; else-if's are possible ala Algol 68: **?? c<sub>1</sub> %t b<sub>1</sub> %e c<sub>2</sub> %t b<sub>2</sub> %e c<sub>3</sub> %t b<sub>3</sub> %e c<sub>4</sub> %t b<sub>4</sub> %e b<sub>5</sub> %;**  
**c<sub>i</sub>** are conditions, **b<sub>i</sub>** are bodies.

If the “-” flag is used with “%[doXs]”, then a colon (:) must be placed between the “%” and the “-” to differentiate the flag from the binary “%-” operator, for example “%:-16.16s”.

Consider the Hewlett-Packard 2645, which, to get to row 3 and column 12, needs to be sent **\E&a12c03Y** padded for 6 milliseconds. Note that the order of the rows and columns is inverted here, and that the row and column are zero-padded as two digits. Thus its **cup** capability is:

**cup=\E&a%p2%2.2dc%p1%2.2dYS<6>**

The Micro-Term ACT-IV needs the current row and column sent preceded by a **^T**, with the row and column simply encoded in binary, “**cup=^T%p1%c%p2%c**”. Devices that use “**%c**” need to be able to backspace the cursor (**cuB1**), and to move the cursor up one line on the screen (**cuU1**). This is necessary because it is not always safe to transmit **\n**, **^D**, and **\r**, as the system may change or discard them. (The library routines dealing with **terminfo** set tty modes so that tabs are never expanded, so **\t** is safe to send. This turns out to be essential for the Ann Arbor 4080.)

A final example is the LSI ADM-3a, which uses row and column offset by a blank character, thus “**cup=\E=%p1%'\s'%+%c%p2%'\s'%+%c**”. After sending “**\E=**”, this pushes the first parameter, pushes the ASCII value for a space (32), adds them (pushing the sum on the stack in place of the two previous values), and outputs that value as a character. Then the same is done for the second parameter. More complex arithmetic is possible using the stack.

### Section 1-3: Cursor Motions

If the terminal has a fast way to home the cursor (to very upper left corner of screen) then this can be given as **home**; similarly a fast way of getting to the lower left-hand corner can be given as **ll**; this may involve going up with **cuU1** from the home position, but a

program should never do this itself (unless **ll** does) because it can make no assumption about the effect of moving up from the home position. Note that the home position is the same as addressing to (0,0): to the top left corner of the screen, not of memory. (Thus, the **\EH** sequence on Hewlett-Packard terminals cannot be used for **home** without losing some of the other features on the terminal.)

If the device has row or column absolute-cursor addressing, these can be given as single parameter capabilities **hpa** (horizontal position absolute) and **vpa** (vertical position absolute). Sometimes these are shorter than the more general two-parameter sequence (as with the Hewlett-Packard 2645) and can be used in preference to **cup**. If there are parameterized local motions (for example, move *n* spaces to the right) these can be given as **cud**, **cub**, **cuf**, and **cuu** with a single parameter indicating how many spaces to move. These are primarily useful if the device does not have **cup**, such as the Tektronix 4025.

If the device needs to be in a special mode when running a program that uses these capabilities, the codes to enter and exit this mode can be given as **smcup** and **rmcup**. This arises, for example, from terminals, such as the Concept, with more than one page of memory. If the device has only memory relative cursor addressing and not screen relative cursor addressing, a one screen-sized window must be fixed into the device for cursor addressing to work properly. This is also used for the Tektronix 4025, where **smcup** sets the command character to be the one used by **terminfo**. If the **smcup** sequence will not restore the screen after an **rmcup** sequence is output (to the state prior to outputting **rmcup**), specify **nrrmc**.

#### Section 1-4: Area Clears

If the terminal can clear from the current position to the end of the line, leaving the cursor where it is, this should be given as **el**. If the terminal can clear from the beginning of the line to the current position inclusive, leaving the cursor where it is, this should be given as **el1**. If the terminal can clear from the current position to the end of the display, then this should be given as **ed**. **ed** is only defined from the first column of a line. (Thus, it can be simulated by a request to delete a large number of lines, if a true **ed** is not available.)

#### Section 1-5: Insert/Delete Line

If the terminal can open a new blank line before the line where the cursor is, this should be given as **il1**; this is done only from the first position of a line. The cursor must then appear on the newly blank line. If the terminal can delete the line which the cursor is on, then this should be given as **dl1**; this is done only from the first position on the line to be deleted. Versions of **il1** and **dl1** which take a single parameter and insert or delete that many lines can be given as **il** and **dl**.

If the terminal has a settable destructive scrolling region (like the VT100) the command to set this can be described with the **csr** capability, which takes two parameters: the top and bottom lines of the scrolling region. The cursor position is, alas, undefined after using this command. It is possible to get the effect of insert or delete line using this command — the **sc** and **rc** (save and restore cursor) commands are also useful. Inserting lines at the

top or bottom of the screen can also be done using **ri** or **ind** on many terminals without a true insert/delete line, and is often faster even on terminals with those features.

To determine whether a terminal has destructive scrolling regions or non-destructive scrolling regions, create a scrolling region in the middle of the screen, place data on the bottom line of the scrolling region, move the cursor to the top line of the scrolling region, and do a reverse index (**ri**) followed by a delete line (**dl1**) or index (**ind**). If the data that was originally on the bottom line of the scrolling region was restored into the scrolling region by the **dl1** or **ind**, then the terminal has non-destructive scrolling regions. Otherwise, it has destructive scrolling regions. Do not specify **csr** if the terminal has non-destructive scrolling regions, unless **ind**, **ri**, **indn**, **rin**, **dl**, and **dl1** all simulate destructive scrolling.

If the terminal has the ability to define a window as part of memory, which all commands affect, it should be given as the parameterized string **wind**. The four parameters are the starting and ending lines in memory and the starting and ending columns in memory, in that order.

If the terminal can retain display memory above, then the **da** capability should be given; if display memory can be retained below, then **db** should be given. These indicate that deleting a line or scrolling a full screen may bring non-blank lines up from below or that scrolling back with **ri** may bring down non-blank lines.

**Section 1-6:**  
**Insert/Delete**  
**Character**

There are two basic kinds of intelligent terminals with respect to insert/delete character operations which can be described using **terminfo**. The most common insert/delete character operations affect only the characters on the current line and shift characters off the end of the line rigidly. Other terminals, such as the Concept 100 and the Perkin Elmer Owl, make a distinction between typed and untyped blanks on the screen, shifting upon an insert or delete only to an untyped blank on the screen which is either eliminated, or expanded to two untyped blanks. You can determine the kind of terminal you have by clearing the screen and then typing text separated by cursor motions. Type “**abc def**” using local cursor motions (not spaces) between the **abc** and the **def**. Then position the cursor before the **abc** and put the terminal in insert mode. If typing characters causes the rest of the line to shift rigidly and characters to fall off the end, then your terminal does not distinguish between blanks and untyped positions. If the **abc** shifts over to the **def** which then move together around the end of the current line and onto the next as you insert, you have the second type of terminal, and should give the capability **in**, which stands for “insert null.” While these are two logically separate attributes (one line versus multiline insert mode, and special treatment of untyped spaces) we have seen no terminals whose insert mode cannot be described with the single attribute.

**terminfo** can describe both terminals that have an insert mode and terminals which send a simple sequence to open a blank position on the current line. Give as **smir** the sequence to get into insert mode. Give as **rmir** the sequence to leave insert mode. Now give as **ich1** any sequence needed to be sent just before sending the character to be inserted.

Most terminals with a true insert mode will not give **ich1**; terminals that send a sequence to open a screen position should give it here. (If your terminal has both, insert mode is usually preferable to **ich1**. Do not give both unless the terminal actually requires both to be used in combination.) If post-insert padding is needed, give this as a number of milliseconds padding in **ip** (a string option). Any other sequence which may need to be sent after an insert of a single character may also be given in **ip**. If your terminal needs both to be placed into an 'insert mode' and a special code to precede each inserted character, then both **smir**/**rmir** and **ich1** can be given, and both will be used. The **ich** capability, with one parameter, *n*, will insert *n* blanks.

If padding is necessary between characters typed while not in insert mode, give this as a number of milliseconds padding in **rmp**.

It is occasionally necessary to move around while in insert mode to delete characters on the same line (for example, if there is a tab after the insertion position). If your terminal allows motion while in insert mode you can give the capability **mir** to speed up inserting in this case. Omitting **mir** will affect only speed. Some terminals (notably Datamedia's) must not have **mir** because of the way their insert mode works.

Finally, you can specify **dch1** to delete a single character, **dch** with one parameter, *n*, to delete *n* characters, and delete mode by giving **smdc** and **rmdc** to enter and exit delete mode (any mode the terminal needs to be placed in for **dch1** to work).

A command to erase *n* characters (equivalent to outputting *n* blanks without moving the cursor) can be given as **ech** with one parameter.

**Section 1-7:  
Highlighting,  
Underlining, and  
Visible Bells**

Your device may have one or more kinds of display attributes that allow you to highlight selected characters when they appear on the screen. The following display modes (shown with the names by which they are set) may be available: a blinking screen (**blink**), bold or extra-bright characters (**bold**), dim or half-bright characters (**dim**), blanking or invisible text (**invis**), protected text (**prot**), a reverse-video screen (**rev**), and an alternate character set (**smacs** to enter this mode and **rmacs** to exit it). (If a command is necessary before you can enter alternate character set mode, give the sequence in **enacs** or "enable alternate-character-set" mode.) Turning on any of these modes singly may or may not turn off other modes.

**sgr0** should be used to turn off all video enhancement capabilities. It should always be specified because it represents the only way to turn off some capabilities, such as **dim** or **blink**.

You should choose one display method as *standout mode* and use it to highlight error messages and other kinds of text to which you want to draw attention. Choose a form of display that provides strong contrast but that is easy on the eyes. (We recommend reverse-video plus half-bright or reverse-video alone.) The sequences to enter and exit standout mode are given as **smso** and **rmso**, respectively. If the code to change into or out of standout mode leaves one or even two blank spaces on the screen, as the TVI 912

and Teleray 1061 do, then **xmc** should be given to tell how many spaces are left.

Sequences to begin underlining and end underlining can be specified as **smul** and **rmul**, respectively. If the device has a sequence to underline the current character and to move the cursor one space to the right (such as the Micro-Term MIME), this sequence can be specified as **uc**.

Terminals with the “magic cookie” glitch (**xmc**) deposit special “cookies” when they receive mode-setting sequences, which affect the display algorithm rather than having extra bits for each character. Some terminals, such as the Hewlett-Packard 2621, automatically leave standout mode when they move to a new line or the cursor is addressed. Programs using standout mode should exit standout mode before moving the cursor or sending a newline, unless the **msg** capability, asserting that it is safe to move in standout mode, is present.

If the terminal has a way of flashing the screen to indicate an error quietly (a bell replacement), then this can be given as **flash**; it must not move the cursor. A good flash can be done by changing the screen into reverse video, pad for 200 ms, then return the screen to normal video.

If the cursor needs to be made more visible than normal when it is not on the bottom line (to make, for example, a non-blinking underline into an easier to find block or blinking underline) give this sequence as **cvvis**. The boolean **chts** should also be given. If there is a way to make the cursor completely invisible, give that as **civis**. The capability **cnorm** should be given which undoes the effects of either of these modes.

If your terminal generates underlined characters by using the underline character (with no special sequences needed) even though it does not otherwise overstrike characters, then you should specify the capability **ul**. For devices on which a character overstriking another leaves both characters on the screen, specify the capability **os**. If overstrikes are erasable with a blank, then this should be indicated by specifying **eo**.

If there is a sequence to set arbitrary combinations of modes, this should be given as **sgr** (set attributes), taking nine parameters. Each parameter is either **0** or non-zero, as the corresponding attribute is on or off. The nine parameters are, in order: standout, underline, reverse, blink, dim, bold, blank, protect, alternate character set. Not all modes need to be supported by **sgr**; only those for which corresponding separate attribute commands exist should be supported. For example, let's assume that the terminal in question needs the following escape sequences to turn on various modes.

tparam parameter	attribute	escape sequence
	none	\E[0m
<b>p1</b>	standout	\E[0;4;7m
<b>p2</b>	underline	\E[0;3m
<b>p3</b>	reverse	\E[0;4m
<b>p4</b>	blink	\E[0;5m
<b>p5</b>	dim	\E[0;7m
<b>p6</b>	bold	\E[0;3;4m
<b>p7</b>	invis	\E[0;8m
<b>p8</b>	protect	not available
<b>p9</b>	altcharset	^O (off) ^N (on)

Note that each escape sequence requires a **0** to turn off other modes before turning on its own mode. Also note that, as suggested above, *standout* is set up to be the combination of *reverse* and *dim*. Also, because this terminal has no *bold* mode, *bold* is set up as the combination of *reverse* and *underline*. In addition, to allow combinations, such as *underline+blink*, the sequence to use would be `\E[0;3;5m`. The terminal doesn't have *protect* mode, either, but that cannot be simulated in any way, so **p8** is ignored. The *altcharset* mode is different in that it is either `^O` or `^N`, depending on whether it is off or on. If all modes were to be turned on, the sequence would be `\E[0;3;4;5;7;8m^N`.

Now look at when different sequences are output. For example, **;3** is output when either **p2** or **p6** is true, that is, if either *underline* or *bold* modes are turned on.

Writing out the above sequences, along with their dependencies, gives the following:

sequence	when to output	terminfo translation
<code>\E[0</code>	always	<code>\E[0</code>
<code>;3</code>	if <b>p2</b> or <b>p6</b>	<code>%%p2%p6%   %t;3%;</code>
<code>;4</code>	if <b>p1</b> or <b>p3</b> or <b>p6</b>	<code>%%p1%p3%   %p6%   %t;4%;</code>
<code>;5</code>	if <b>p4</b>	<code>%%p4%t;5%;</code>
<code>;7</code>	if <b>p1</b> or <b>p5</b>	<code>%%p1%p5%   %t;7%;</code>
<code>;8</code>	if <b>p7</b>	<code>%%p7%t;8%;</code>
<b>m</b>	always	<b>m</b>
<code>^N</code> or <code>^O</code>	if <b>p9</b> ^N, else ^O	<code>%%p9%t^N%e^O%;</code>

Putting this all together into the **sgr** sequence gives:

```
sgr=\E[0%%p2%p6% | %t;3%;%%p1%p3% | %p6%
| %t;4%;%%p5%t;5%;%%p1%p5%
| %t;7%;%%p7%t;8%;m%%p9%t^N%e^O%;
```

Remember that **sgr** and **sgr0** must always be specified.

### Section 1-8: Keypad

If the device has a keypad that transmits sequences when the keys are pressed, this information can also be specified. Note that it is not possible to handle devices where the keypad only works in local (this applies, for example, to the unshifted Hewlett-Packard

2621 keys). If the keypad can be set to transmit or not transmit, specify these sequences as **smkx** and **rmkx**. Otherwise the keypad is assumed to always transmit.

The sequences sent by the left arrow, right arrow, up arrow, down arrow, and home keys can be given as **kcub1**, **kcuf1**, **kcuu1**, **kcud1**, and **khome**, respectively. If there are function keys such as f0, f1, ..., f63, the sequences they send can be specified as **kf0**, **kf1**, ..., **kf63**. If the first 11 keys have labels other than the default f0 through f10, the labels can be given as **lf0**, **lf1**, ..., **lf10**. The codes transmitted by certain other special keys can be given: **kll** (home down), **kbs** (backspace), **ktbc** (clear all tabs), **kctab** (clear the tab stop in this column), **kclr** (clear screen or erase key), **kdch1** (delete character), **kdl1** (delete line), **krmir** (exit insert mode), **kel** (clear to end of line), **ked** (clear to end of screen), **kich1** (insert character or enter insert mode), **kil1** (insert line), **knp** (next page), **kpp** (previous page), **kind** (scroll forward/down), **kri** (scroll backward/up), **khts** (set a tab stop in this column). In addition, if the keypad has a 3 by 3 array of keys including the four arrow keys, the other five keys can be given as **ka1**, **ka3**, **kb2**, **kc1**, and **kc3**. These keys are useful when the effects of a 3 by 3 directional pad are needed. Further keys are defined above in the capabilities list.

Strings to program function keys can be specified as **pfkey**, **pfloc**, and **px**. A string to program screen labels should be specified as **pln**. Each of these strings takes two parameters: a function key identifier and a string to program it with. **pfkey** causes pressing the given key to be the same as the user typing the given string; **pfloc** causes the string to be executed by the terminal in local mode; and **px** causes the string to be transmitted to the computer. The capabilities **nlab**, **lw** and **lh** define the number of programmable screen labels and their width and height. If there are commands to turn the labels on and off, give them in **smln** and **rmln**. **smln** is normally output after one or more **pln** sequences to make sure that the change becomes visible.

### Section 1-9: Tabs and Initialization

If the device has hardware tabs, the command to advance to the next tab stop can be given as **ht** (usually control I). A "backtab" command that moves leftward to the next tab stop can be given as **cbt**. By convention, if tty modes show that tabs are being expanded by the computer rather than being sent to the device, programs should not use **ht** or **cbt** (even if they are present) because the user may not have the tab stops properly set. If the device has hardware tabs that are initially set every *n* spaces when the device is powered up, the numeric parameter **it** is given, showing the number of spaces the tabs are set to. This is normally used by **tput init** (see **tput(1)**) to determine whether to set the mode for hardware tab expansion and whether to set the tab stops. If the device has tab stops that can be saved in nonvolatile memory, the **terminfo** description can assume that they are properly set. If there are commands to set and clear tab stops, they can be given as **tbc** (clear all tab stops) and **hts** (set a tab stop in the current column of every row).

Other capabilities include: **is1**, **is2**, and **is3**, initialization strings for the device; **iprogr**, the path name of a program to be run to initialize the device; and **if**, the name of a file containing long initialization strings. These strings are expected to set the device into modes

consistent with the rest of the **terminfo** description. They must be sent to the device each time the user logs in and be output in the following order: run the program **iprogram**; output **is1**; output **is2**; set the margins using **mgc**, **smgl** and **smgr**; set the tabs using **tbc** and **hts**; print the file **if**; and finally output **is3**. This is usually done using the **init** option of **tput**.

Most initialization is done with **is2**. Special device modes can be set up without duplicating strings by putting the common sequences in **is2** and special cases in **is1** and **is3**.

Sequences that do a reset from a totally unknown state can be given as **rs1**, **rs2**, **rf**, and **rs3**, analogous to **is1**, **is2**, **is3**, and **if**. (The method using files, **if** and **rf**, is used for a few terminals, from **/usr/share/lib/tabset/\***; however, the recommended method is to use the initialization and reset strings.) These strings are output by **tput** **reset**, which is used when the terminal gets into a wedged state. Commands are normally placed in **rs1**, **rs2**, **rs3**, and **rf** only if they produce annoying effects on the screen and are not necessary when logging in. For example, the command to set a terminal into 80-column mode would normally be part of **is2**, but on some terminals it causes an annoying glitch on the screen and is not normally needed because the terminal is usually already in 80-column mode.

If a more complex sequence is needed to set the tabs than can be described by using **tbc** and **hts**, the sequence can be placed in **is2** or **if**.

Any margin can be cleared with **mgc**. (For instructions on how to specify commands to set and clear margins, see "Margins" below under "PRINTER CAPABILITIES.")

#### Section 1-10: Delays

Certain capabilities control padding in the **tty** driver. These are primarily needed by hard-copy terminals, and are used by **tput** **init** to set **tty** modes appropriately. Delays embedded in the capabilities **cr**, **ind**, **cub1**, **ff**, and **tab** can be used to set the appropriate delay bits to be set in the **tty** driver. If **pb** (padding baud rate) is given, these values can be ignored at baud rates below the value of **pb**.

#### Section 1-11: Status Lines

If the terminal has an extra "status line" that is not normally used by software, this fact can be indicated. If the status line is viewed as an extra line below the bottom line, into which one can cursor address normally (such as the Heathkit h19's 25th line, or the 24th line of a VT100 which is set to a 23-line scrolling region), the capability **hs** should be given. Special strings that go to a given column of the status line and return from the status line can be given as **tsl** and **fsl**. (**fsl** must leave the cursor position in the same place it was before **tsl**. If necessary, the **sc** and **rc** strings can be included in **tsl** and **fsl** to get this effect.) The capability **tsl** takes one parameter, which is the column number of the status line the cursor is to be moved to.

If escape sequences and other special commands, such as **tab**, work while in the status line, the flag **eslok** can be given. A string which turns off the status line (or otherwise erases its contents) should be given as **dsl**. If the terminal has commands to save and restore the position of the cursor, give them as **sc** and **rc**. The status line is normally assumed to be the same width as the rest of the screen, for example, **cols**. If the status



**Section 1-12: Line Graphics**

line is a different width (possibly because the terminal does not allow an entire line to be loaded) the width, in columns, can be indicated with the numeric parameter **wsl**.

If the device has a line drawing alternate character set, the mapping of glyph to character would be given in **acsc**. The definition of this string is based on the alternate character set used in the DEC VT100 terminal, extended slightly with some characters from the AT&T 4410v1 terminal.

glyph name	vt100+ character
arrow pointing right	+
arrow pointing left	,
arrow pointing down	.
solid square block	<b>0</b>
lantern symbol	<b>I</b>
arrow pointing up	-
diamond	'
checker board (stipple)	<b>a</b>
degree symbol	<b>f</b>
plus/minus	<b>g</b>
board of squares	<b>h</b>
lower right corner	<b>j</b>
upper right corner	<b>k</b>
upper left corner	<b>l</b>
lower left corner	<b>m</b>
plus	<b>n</b>
scan line 1	<b>o</b>
horizontal line	<b>q</b>
scan line 9	<b>s</b>
left tee (┌)	<b>t</b>
right tee (┐)	<b>u</b>
bottom tee (└)	<b>v</b>
top tee (┘)	<b>w</b>
vertical line	<b>x</b>
bullet	~

The best way to describe a new device's line graphics set is to add a third column to the above table with the characters for the new device that produce the appropriate glyph when the device is in the alternate character set mode. For example,

glyph name	vt100+ char	new tty char
upper left corner	<b>l</b>	<b>R</b>
lower left corner	<b>m</b>	<b>F</b>
upper right corner	<b>k</b>	<b>T</b>
lower right corner	<b>j</b>	<b>G</b>
horizontal line	<b>q</b>	,
vertical line	<b>x</b>	.

Now write down the characters left to right, as in “**acsc=lRmFkTjGq\,x.**”.

In addition, **terminfo** allows you to define multiple character sets. See Section 2-5 for details.

### Section 1-13: Color Manipulation

Let us define two methods of color manipulation: the Tektronix method and the HP method. The Tektronix method uses a set of  $N$  predefined colors (usually 8) from which a user can select "current" foreground and background colors. Thus a terminal can support up to  $N$  colors mixed into  $N*N$  color-pairs to be displayed on the screen at the same time. When using an HP method the user cannot define the foreground independently of the background, or vice-versa. Instead, the user must define an entire color-pair at once. Up to  $M$  color-pairs, made from  $2*M$  different colors, can be defined this way. Most existing color terminals belong to one of these two classes of terminals.

The numeric variables **colors** and **pairs** define the number of colors and color-pairs that can be displayed on the screen at the same time. If a terminal can change the definition of a color (for example, the Tektronix 4100 and 4200 series terminals), this should be specified with **ccc** (can change color). To change the definition of a color (Tektronix 4200 method), use **initc** (initialize color). It requires four arguments: color number (ranging from 0 to **colors**-1) and three RGB (red, green, and blue) values or three HLS colors (Hue, Lightness, Saturation). Ranges of RGB and HLS values are terminal dependent.

Tektronix 4100 series terminals only use HLS color notation. For such terminals (or dual-mode terminals to be operated in HLS mode) one must define a boolean variable **hls**; that would instruct the **courses init\_color** routine to convert its RGB arguments to HLS before sending them to the terminal. The last three arguments to the **initc** string would then be HLS values.

If a terminal can change the definitions of colors, but uses a color notation different from RGB and HLS, a mapping to either RGB or HLS must be developed.

To set current foreground or background to a given color, use **setaf** (set ANSI foreground) and **setab** (set ANSI background). They require one parameter: the number of the color. To initialize a color-pair (HP method), use **initp** (initialize pair). It requires

seven parameters: the number of a color-pair (range=0 to **pairs**-1), and six RGB values: three for the foreground followed by three for the background. (Each of these groups of three should be in the order RGB.) When **initc** or **initp** are used, RGB or HLS arguments should be in the order "red, green, blue" or "hue, lightness, saturation", respectively. To make a color-pair current, use **scp** (set color-pair). It takes one parameter, the number of a color-pair.

Some terminals (for example, most color terminal emulators for PCs) erase areas of the screen with current background color. In such cases, **bce** (background color erase) should be defined. The variable **op** (original pair) contains a sequence for setting the foreground and the background colors to what they were at the terminal start-up time. Similarly, **oc** (original colors) contains a control sequence for setting all colors (for the Tektronix method) or color-pairs (for the HP method) to the values they had at the terminal start-up time.

Some color terminals substitute color for video attributes. Such video attributes should not be combined with colors. Information about these video attributes should be packed into the **ncv** (no color video) variable. There is a one-to-one correspondence between the nine least significant bits of that variable and the video attributes. The following table depicts this correspondence.

Attribute	Bit Position	Decimal Value
<b>A_STANDOUT</b>	0	1
<b>A_UNDERLINE</b>	1	2
<b>A_REVERSE</b>	2	4
<b>A_BLINK</b>	3	8
<b>A_DIM</b>	4	16
<b>A_BOLD</b>	5	32
<b>A_INVIS</b>	6	64
<b>A_PROTECT</b>	7	128
<b>A_ALTCHARSET</b>	8	256

When a particular video attribute should not be used with colors, the corresponding **ncv** bit should be set to 1; otherwise it should be set to zero. To determine the information to pack into the **ncv** variable, you must add together the decimal values corresponding to those attributes that cannot coexist with colors. For example, if the terminal uses colors to simulate reverse video (bit number 2 and decimal value 4) and bold (bit number 5 and decimal value 32), the resulting value for **ncv** will be 36 (4 + 32).

**Section 1-14:**  
**Miscellaneous**

If the terminal requires other than a null (zero) character as a pad, then this can be given as **pad**. Only the first character of the **pad** string is used. If the terminal does not have a pad character, specify **npc**.

If the terminal can move up or down half a line, this can be indicated with **hu** (half-line up) and **hd** (half-line down). This is primarily useful for superscripts and subscripts on hardcopy terminals. If a hardcopy terminal can eject to the next page (form feed), give this as **ff** (usually control L).

If there is a command to repeat a given character a given number of times (to save time transmitting a large number of identical characters) this can be indicated with the parameterized string **rep**. The first parameter is the character to be repeated and the second is the number of times to repeat it. Thus, **tparam(repeat\_char, 'x', 10)** is the same as **xxxxxxxxxx**.

If the terminal has a settable command character, such as the Tektronix 4025, this can be indicated with **cmdch**. A prototype command character is chosen which is used in all capabilities. This character is given in the **cmdch** capability to identify it. The following convention is supported on some systems: If the environment variable **CC** exists, all occurrences of the prototype character are replaced with the character in **CC**.

Terminal descriptions that do not represent a specific kind of known terminal, such as *switch*, *dialup*, *patch*, and *network*, should include the **gn** (generic) capability so that programs can complain that they do not know how to talk to the terminal. (This capability does not apply to *virtual* terminal descriptions for which the escape sequences are known.) If the terminal is one of those supported by the system virtual terminal protocol, the terminal number can be given as **vt**. A line-turn-around sequence to be transmitted before doing reads should be specified in **rft**.

If the device uses xon/xoff handshaking for flow control, give **xon**. Padding information should still be included so that routines can make better decisions about costs, but actual pad characters will not be transmitted. Sequences to turn on and off xon/xoff handshaking may be given in **smxon** and **rmxon**. If the characters used for handshaking are not **^S** and **^Q**, they may be specified with **xonc** and **xoffc**.

If the terminal has a “meta key” which acts as a shift key, setting the 8th bit of any character transmitted, this fact can be indicated with **km**. Otherwise, software will assume that the 8th bit is parity and it will usually be cleared. If strings exist to turn this “meta mode” on and off, they can be given as **smm** and **rmm**.

If the terminal has more lines of memory than will fit on the screen at once, the number of lines of memory can be indicated with **lm**. A value of **lm#0** indicates that the number of lines is not fixed, but that there is still more memory than fits on the screen.

Media copy strings which control an auxiliary printer connected to the terminal can be given as **mc0**: print the contents of the screen, **mc4**: turn off the printer, and **mc5**: turn on the printer. When the printer is on, all text sent to the terminal will be sent to the printer.

### Section 1-15: Special Cases

A variation, **mc5p**, takes one parameter, and leaves the printer on for as many characters as the value of the parameter, then turns the printer off. The parameter should not exceed 255. If the text is not displayed on the terminal screen when the printer is on, specify **mc5i** (silent printer). All text, including **mc4**, is transparently passed to the printer while an **mc5p** is in effect.

The working model used by **terminfo** fits most terminals reasonably well. However, some terminals do not completely match that model, requiring special support by **terminfo**. These are not meant to be construed as deficiencies in the terminals; they are just differences between the working model and the actual hardware. They may be unusual devices or, for some reason, do not have all the features of the **terminfo** model implemented.

Terminals that cannot display tilde (~) characters, such as certain Hazeltine terminals, should indicate **hz**.

Terminals that ignore a linefeed immediately after an **am** wrap, such as the Concept 100, should indicate **xenl**. Those terminals whose cursor remains on the right-most column until another character has been received, rather than wrapping immediately upon receiving the right-most character, such as the VT100, should also indicate **xenl**.

If **el** is required to get rid of standout (instead of writing normal text on top of it), **xhp** should be given.

Those Teleray terminals whose tabs turn all characters moved over to blanks, should indicate **xt** (destructive tabs). This capability is also taken to mean that it is not possible to position the cursor on top of a “magic cookie.” Therefore, to erase standout mode, it is necessary, instead, to use delete and insert line.

Those Beehive Superbee terminals which do not transmit the escape or control-C characters, should specify **xsb**, indicating that the f1 key is to be used for escape and the f2 key for control C.

### Section 1-16: Similar Terminals

If there are two very similar terminals, one can be defined as being just like the other with certain exceptions. The string capability **use** can be given with the name of the similar terminal. The capabilities given before **use** override those in the terminal type invoked by **use**. A capability can be canceled by placing **xx@** to the left of the capability definition, where **xx** is the capability. For example, the entry

```
att4424-2 | Teletype 4424 in display function group ii,  
rev@, sgr@, smul@, use=att4424,
```

defines an AT&T 4424 terminal that does not have the **rev**, **sgr**, and **smul** capabilities, and hence cannot do highlighting. This is useful for different modes for a terminal, or for different user preferences. More than one **use** capability may be given.

**PART 2: PRINTER  
CAPABILITIES**

The **terminfo** database allows you to define capabilities of printers as well as terminals. To find out what capabilities are available for printers as well as for terminals, see the two lists under "DEVICE CAPABILITIES" that list capabilities by variable and by capability name.

**Section 2-1:  
Rounding Values**

Because parameterized string capabilities work only with integer values, we recommend that **terminfo** designers create strings that expect numeric values that have been rounded. Application designers should note this and should always round values to the nearest integer before using them with a parameterized string capability.

**Section 2-2: Printer  
Resolution**

A printer's resolution is defined to be the smallest spacing of characters it can achieve. In general printers have independent resolution horizontally and vertically. Thus the vertical resolution of a printer can be determined by measuring the smallest achievable distance between consecutive printing baselines, while the horizontal resolution can be determined by measuring the smallest achievable distance between the left-most edges of consecutive printed, identical, characters.

All printers are assumed to be capable of printing with a uniform horizontal and vertical resolution. The view of printing that **terminfo** currently presents is one of printing inside a uniform matrix: All characters are printed at fixed positions relative to each "cell" in the matrix; furthermore, each cell has the same size given by the smallest horizontal and vertical step sizes dictated by the resolution. (The cell size can be changed as will be seen later.)

Many printers are capable of "proportional printing," where the horizontal spacing depends on the size of the character last printed. **terminfo** does not make use of this capability, although it does provide enough capability definitions to allow an application to simulate proportional printing.

A printer must not only be able to print characters as close together as the horizontal and vertical resolutions suggest, but also of "moving" to a position an integral multiple of the smallest distance away from a previous position. Thus printed characters can be spaced apart a distance that is an integral multiple of the smallest distance, up to the length or width of a single page.

Some printers can have different resolutions depending on different "modes." In "normal mode," the existing **terminfo** capabilities are assumed to work on columns and lines, just like a video terminal. Thus the old **lines** capability would give the length of a page in lines, and the **cols** capability would give the width of a page in columns. In "micro mode," many **terminfo** capabilities work on increments of lines and columns. With some printers the micro mode may be concomitant with normal mode, so that all the capabilities work at the same time.

**Section 2-3:  
Specifying Printer  
Resolution**

The printing resolution of a printer is given in several ways. Each specifies the resolution as the number of smallest steps per distance:

Specification of Printer Resolution  
Characteristic Number of Smallest Steps

<b>orhi</b>	Steps per inch horizontally
<b>orvi</b>	Steps per inch vertically
<b>orc</b>	Steps per column
<b>orl</b>	Steps per line

When printing in normal mode, each character printed causes movement to the next column, except in special cases described later; the distance moved is the same as the per-column resolution. Some printers cause an automatic movement to the next line when a character is printed in the rightmost position; the distance moved vertically is the same as the per-line resolution.

When printing in micro mode, these distances can be different, and may be zero for some printers.

Specification of Printer Resolution  
Automatic Motion after Printing

*Normal Mode:*

<b>orc</b>	Steps moved horizontally
<b>orl</b>	Steps moved vertically

*Micro Mode:*

<b>mcs</b>	Steps moved horizontally
<b>mls</b>	Steps moved vertically

Some printers are capable of printing wide characters. The distance moved when a wide character is printed in normal mode may be different from when a regular width character is printed. The distance moved when a wide character is printed in micro mode may also be different from when a regular character is printed in micro mode, but the differences are assumed to be related: If the distance moved for a regular character is the same whether in normal mode or micro mode (**mcs=orc**), then the distance moved for a wide character is also the same whether in normal mode or micro mode. This doesn't mean the normal character distance is necessarily the same as the wide character distance, just that the distances don't change with a change in normal to micro mode. However, if the distance moved for a regular character is different in micro mode from the distance moved in normal mode (**mcs<orc**), the micro mode distance is assumed to be the same for a wide character printed in micro mode, as the table below shows.

Specification of Printer Resolution  
Automatic Motion after Printing Wide Character

---

*Normal Mode or Micro Mode (mcs = orc):*

**widcs** Steps moved horizontally

*Micro Mode (mcs < orc):*

**mcs** Steps moved horizontally

There may be control sequences to change the number of columns per inch (the character pitch) and to change the number of lines per inch (the line pitch). If these are used, the resolution of the printer changes, but the type of change depends on the printer:

Specification of Printer Resolution  
Changing the Character/Line Pitches

---

<b>cpi</b>	Change character pitch
<b>cpix</b>	If set, <b>cpi</b> changes <b>orhi</b> , otherwise changes <b>orc</b>
<b>lpi</b>	Change line pitch
<b>lpix</b>	If set, <b>lpi</b> changes <b>orvi</b> , otherwise changes <b>orl</b>
<b>chr</b>	Change steps per column
<b>cvr</b>	Change steps per line

The **cpi** and **lpi** string capabilities are each used with a single argument, the pitch in columns (or characters) and lines per inch, respectively. The **chr** and **cvr** string capabilities are each used with a single argument, the number of steps per column and line, respectively.

Using any of the control sequences in these strings will imply a change in some of the values of **orc**, **orhi**, **orl**, and **orvi**. Also, the distance moved when a wide character is printed, **widcs**, changes in relation to **orc**. The distance moved when a character is printed in micro mode, **mcs**, changes similarly, with one exception: if the distance is 0 or 1, then no change is assumed (see items marked with † in the following table).

Programs that use **cpi**, **lpi**, **chr**, or **cvr** should recalculate the printer resolution (and should recalculate other values— see "Effect of Changing Printing Resolution" under "Dot-Mapped Graphics").



Specification of Printer Resolution  
Effects of Changing the Character/Line Pitches

<i>Before</i>	<i>After</i>
Using <b>cpi</b> with <b>cpix</b> clear:	
$\$bold\ orhi\ '\$$	<b>orhi</b>
$\$bold\ orc\ '\$$	$\$bold\ orc = bold\ orhi\ over\ V\ sub\ italic\ cpi\$$
Using <b>cpi</b> with <b>cpix</b> set:	
$\$bold\ orhi\ '\$$	$\$bold\ orhi = bold\ orc\ cdot\ V\ sub\ italic\ cpi\$$
$\$bold\ orc\ '\$$	$\$bold\ orc\$$
Using <b>lpi</b> with <b>lpix</b> clear:	
$\$bold\ orvi\ '\$$	$\$bold\ orvi\$$
$\$bold\ orl\ '\$$	$\$bold\ orl = bold\ orvi\ over\ V\ sub\ italic\ lpi\$$
Using <b>lpi</b> with <b>lpix</b> set:	
$\$bold\ orvi\ '\$$	$\$bold\ orvi = bold\ orl\ cdot\ V\ sub\ italic\ lpi\$$
$\$bold\ orl\ '\$$	$\$bold\ orl\$$
Using <b>chr</b> :	
$\$bold\ orhi\ '\$$	$\$bold\ orhi\$$
$\$bold\ orc\ '\$$	$\$V\ sub\ italic\ chr\$$
Using <b>cvr</b> :	
$\$bold\ orvi\ '\$$	$\$bold\ orvi\$$
$\$bold\ orl\ '\$$	$\$V\ sub\ italic\ cvr\$$
Using <b>cpi</b> or <b>chr</b> :	
$\$bold\ widcs\ '\$$	$\$bold\ widcs = bold\ \{widcs\ '\}\ bold\ orc\ over\ \{bold\ \{orc\ '\}\}\$$
$\$bold\ mcs\ '\$$	$\$bold\ mcs = bold\ \{mcs\ '\}\ bold\ orc\ over\ \{bold\ \{orc\ '\}\}\$$

$\$V\ sub\ italic\ cpi\$$ ,  $\$V\ sub\ italic\ lpi\$$ ,  $\$V\ sub\ italic\ chr\$$ , and  $\$V\ sub\ italic\ cvr\$$  are the arguments used with **cpi**, **lpi**, **chr**, and **cvr**, respectively. The prime marks ( ' ) indicate the old values.

**Section 2-4:**  
**Capabilities that**  
**Cause Movement**

In the following descriptions, “movement” refers to the motion of the “current position.” With video terminals this would be the cursor; with some printers this is the carriage position. Other printers have different equivalents. In general, the current position is where a character would be displayed if printed.

**terminfo** has string capabilities for control sequences that cause movement a number of full columns or lines. It also has equivalent string capabilities for control sequences that cause movement a number of smallest steps.

---

String Capabilities for Motion

---

<b>mcub1</b>	Move 1 step left
<b>mcuf1</b>	Move 1 step right
<b>mcuu1</b>	Move 1 step up
<b>mcud1</b>	Move 1 step down
<b>mcub</b>	Move <i>N</i> steps left
<b>mcuf</b>	Move <i>N</i> steps right
<b>mcuu</b>	Move <i>N</i> steps up
<b>mcud</b>	Move <i>N</i> steps down
<b>mhpa</b>	Move <i>N</i> steps from the left
<b>mvpa</b>	Move <i>N</i> steps from the top

The latter six strings are each used with a single argument, *N*.

Sometimes the motion is limited to less than the width or length of a page. Also, some printers don't accept absolute motion to the left of the current position. **terminfo** has capabilities for specifying these limits.

---

Limits to Motion

---

<b>mjump</b>	Limit on use of <b>mcub1</b> , <b>mcuf1</b> , <b>mcuu1</b> , <b>mcud1</b>
<b>maddr</b>	Limit on use of <b>mhpa</b> , <b>mvpa</b>
<b>xhpa</b>	If set, <b>hpa</b> and <b>mhpa</b> can't move left
<b>xvpa</b>	If set, <b>vpa</b> and <b>mvpa</b> can't move up

If a printer needs to be in a "micro mode" for the motion capabilities described above to work, there are string capabilities defined to contain the control sequence to enter and exit this mode. A boolean is available for those printers where using a carriage return causes an automatic return to normal mode.

---

Entering/Exiting Micro Mode

---

<b>smicm</b>	Enter micro mode
<b>rmicm</b>	Exit micro mode
<b>crxm</b>	Using <b>cr</b> exits micro mode

The movement made when a character is printed in the rightmost position varies among printers. Some make no movement, some move to the beginning of the next line, others move to the beginning of the same line. **terminfo** has boolean capabilities for describing all three cases.

---

What Happens After Character  
Printed in Rightmost Position

---

<b>sam</b>	Automatic move to beginning of same line
------------	--

Some printers can be put in a mode where the normal direction of motion is reversed. This mode can be especially useful when there are no capabilities for leftward or upward motion, because those capabilities can be built from the motion reversal capability and the rightward or downward motion capabilities. It is best to leave it up to an application

to build the leftward or upward capabilities, though, and not enter them in the **terminfo** database. This allows several reverse motions to be strung together without intervening wasted steps that leave and reenter reverse mode.

Entering/Exiting Reverse Modes

<b>slm</b>	Reverse sense of horizontal motions
<b>rlm</b>	Restore sense of horizontal motions
<b>sum</b>	Reverse sense of vertical motions
<b>rum</b>	Restore sense of vertical motions

*While sense of horizontal motions reversed:*

<b>mcub1</b>	Move 1 step right
<b>mcuf1</b>	Move 1 step left
<b>mcub</b>	Move <i>N</i> steps right
<b>mcuf</b>	Move <i>N</i> steps left
<b>cub1</b>	Move 1 column right
<b>cuf1</b>	Move 1 column left
<b>cub</b>	Move <i>N</i> columns right
<b>cuf</b>	Move <i>N</i> columns left

*While sense of vertical motions reversed:*

<b>mcuu1</b>	Move 1 step down
<b>mcud1</b>	Move 1 step up
<b>mcuu</b>	Move <i>N</i> steps down
<b>mcud</b>	Move <i>N</i> steps up
<b>cuu1</b>	Move 1 line down
<b>cud1</b>	Move 1 line up
<b>cuu</b>	Move <i>N</i> lines down
<b>cud</b>	Move <i>N</i> lines up

The reverse motion modes should not affect the **mvpa** and **mhpa** absolute motion capabilities. The reverse vertical motion mode should, however, also reverse the action of the line “wrapping” that occurs when a character is printed in the right-most position. Thus printers that have the standard **terminfo** capability **am** defined should experience motion to the beginning of the previous line when a character is printed in the right-most position under reverse vertical motion mode.

The action when any other motion capabilities are used in reverse motion modes is not defined; thus, programs must exit reverse motion modes before using other motion capabilities.

Two miscellaneous capabilities complete the list of new motion capabilities. One of these is needed for printers that move the current position to the beginning of a line when certain control characters, such as “line-feed” or “form-feed,” are used. The other is used for the capability of suspending the motion that normally occurs after printing a character.

---

 Miscellaneous Motion Strings
 

---

<b>docr</b>	List of control characters causing <b>cr</b>
<b>zerom</b>	Prevent auto motion after printing next single character

## Margins

**terminfo** provides two strings for setting margins on terminals: one for the left and one for the right margin. Printers, however, have two additional margins, for the top and bottom margins of each page. Furthermore, some printers require not using motion strings to move the current position to a margin and then fixing the margin there, but require the specification of where a margin should be regardless of the current position. Therefore **terminfo** offers six additional strings for defining margins with printers.

---

 Setting Margins
 

---

<b>smgl</b>	Set left margin at current column
<b>smgr</b>	Set right margin at current column
<b>smgb</b>	Set bottom margin at current line
<b>smgt</b>	Set top margin at current line
<b>smgbp</b>	Set bottom margin at line <i>N</i>
<b>smglp</b>	Set left margin at column <i>N</i>
<b>smgrp</b>	Set right margin at column <i>N</i>
<b>smgtp</b>	Set top margin at line <i>N</i>

The last four strings are used with one or more arguments that give the position of the margin or margins to set. If both of **smglp** and **smgrp** are set, each is used with a single argument, *N*, that gives the column number of the left and right margin, respectively. If both of **smgtp** and **smgbp** are set, each is used to set the top and bottom margin, respectively: **smgtp** is used with a single argument, *N*, the line number of the top margin; however, **smgbp** is used with two arguments, *N* and *M*, that give the line number of the bottom margin, the first counting from the top of the page and the second counting from the bottom. This accommodates the two styles of specifying the bottom margin in different manufacturers' printers. When coding a **terminfo** entry for a printer that has a settable bottom margin, only the first or second parameter should be used, depending on the printer. When writing an application that uses **smgbp** to set the bottom margin, both arguments must be given.

If only one of **smglp** and **smgrp** is set, then it is used with two arguments, the column number of the left and right margins, in that order. Likewise, if only one of **smgtp** and **smgbp** is set, then it is used with two arguments that give the top and bottom margins, in that order, counting from the top of the page. Thus when coding a **terminfo** entry for a printer that requires setting both left and right or top and bottom margins simultaneously, only one of **smglp** and **smgrp** or **smgtp** and **smgbp** should be defined; the other should be left blank. When writing an application that uses these string capabilities, the pairs should be first checked to see if each in the pair is set or only one is set, and should then be used accordingly.

**Shadows, Italics,  
Wide Characters**

In counting lines or columns, line zero is the top line and column zero is the left-most column. A zero value for the second argument with **smgpb** means the bottom line of the page.

All margins can be cleared with **mgc**.

Five new sets of strings describe the capabilities printers have of enhancing printed text.

Enhanced Printing

---

<b>sshm</b>	Enter shadow-printing mode
<b>rshm</b>	Exit shadow-printing mode
<b>sitm</b>	Enter italicizing mode
<b>ritm</b>	Exit italicizing mode
<b>swidm</b>	Enter wide character mode
<b>rwidm</b>	Exit wide character mode
<b>ssupm</b>	Enter superscript mode
<b>rsupm</b>	Exit superscript mode
<b>supcs</b>	List of characters available as superscripts
<b>ssubm</b>	Enter subscript mode
<b>rsubm</b>	Exit subscript mode
<b>subcs</b>	List of characters available as subscripts

If a printer requires the **sshm** control sequence before every character to be shadow-printed, the **rshm** string is left blank. Thus programs that find a control sequence in **sshm** but none in **rshm** should use the **sshm** control sequence before every character to be shadow-printed; otherwise, the **sshm** control sequence should be used once before the set of characters to be shadow-printed, followed by **rshm**. The same is also true of each of the **sitm/ritm**, **swidm/rwidm**, **ssupm/rsupm**, and **ssubm/rsubm** pairs.

Note that **terminfo** also has a capability for printing emboldened text (**bold**). While shadow printing and emboldened printing are similar in that they “darken” the text, many printers produce these two types of print in slightly different ways. Generally, emboldened printing is done by overstriking the same character one or more times. Shadow printing likewise usually involves overstriking, but with a slight movement up and/or to the side so that the character is “fatter.”

It is assumed that enhanced printing modes are independent modes, so that it would be possible, for instance, to shadow print italicized subscripts.

As mentioned earlier, the amount of motion automatically made after printing a wide character should be given in **widcs**.

If only a subset of the printable ASCII characters can be printed as superscripts or subscripts, they should be listed in **supcs** or **subcs** strings, respectively. If the **ssupm** or **ssubm** strings contain control sequences, but the corresponding **supcs** or **subcs** strings

are empty, it is assumed that all printable ASCII characters are available as superscripts or subscripts.

Automatic motion made after printing a superscript or subscript is assumed to be the same as for regular characters. Thus, for example, printing any of the following three examples will result in equivalent motion:

**Bi B<sub>i</sub> B<sup>i</sup>**

Note that the existing **msgr** boolean capability describes whether motion control sequences can be used while in “standout mode.” This capability is extended to cover the enhanced printing modes added here. **msgr** should be set for those printers that accept any motion control sequences without affecting shadow, italicized, widened, superscript, or subscript printing. Conversely, if **msgr** is not set, a program should end these modes before attempting any motion.

### Section 2-5: Alternate Character Sets

In addition to allowing you to define line graphics (described in Section 1-12), **terminfo** lets you define alternate character sets. The following capabilities cover printers and terminals with multiple selectable or definable character sets.

#### Alternate Character Sets

<b>scs</b>	Select character set <i>N</i>
<b>scsd</b>	Start definition of character set <i>N</i> , <i>M</i> characters
<b>defc</b>	Define character <i>A</i> , <i>B</i> dots wide, descender <i>D</i>
<b>rcsd</b>	End definition of character set <i>N</i>
<b>csnm</b>	List of character set names
<b>daisy</b>	Printer has manually changed print-wheels

The **scs**, **rcsd**, and **csnm** strings are used with a single argument, *N*, a number from 0 to 63 that identifies the character set. The **scsd** string is also used with the argument *N* and another, *M*, that gives the number of characters in the set. The **defc** string is used with three arguments: *A* gives the ASCII code representation for the character, *B* gives the width of the character in dots, and *D* is zero or one depending on whether the character is a “descender” or not. The **defc** string is also followed by a string of “image-data” bytes that describe how the character looks (see below).

Character set 0 is the default character set present after the printer has been initialized. Not every printer has 64 character sets, of course; using **scs** with an argument that doesn’t select an available character set should cause a null result from **tparm**.

If a character set has to be defined before it can be used, the **scsd** control sequence is to be used before defining the character set, and the **rcsd** is to be used after. They should also cause a null result from **tparm** when used with an argument *N* that doesn’t apply. If a character set still has to be selected after being defined, the **scs** control sequence should follow the **rcsd** control sequence. By examining the results of using each of the **scs**, **scsd**,

and **rcsd** strings with a character set number in a call to **tparm**, a program can determine which of the three are needed.

Between use of the **sbsd** and **rcsd** strings, the **defc** string should be used to define each character. To print any character on printers covered by **terminfo**, the ASCII code is sent to the printer. This is true for characters in an alternate set as well as “normal” characters. Thus the definition of a character includes the ASCII code that represents it. In addition, the width of the character in dots is given, along with an indication of whether the character should descend below the print line (such as the lower case letter “g” in most character sets). The width of the character in dots also indicates the number of image-data bytes that will follow the **defc** string. These image-data bytes indicate where in a dot-matrix pattern ink should be applied to “draw” the character; the number of these bytes and their form are defined below under “Dot-Mapped Graphics.”

It’s easiest for the creator of **terminfo** entries to refer to each character set by number; however, these numbers will be meaningless to the application developer. The **csnm** string alleviates this problem by providing names for each number.

When used with a character set number in a call to **tparm**, the **csnm** string will produce the equivalent name. These names should be used as a reference only. No naming convention is implied, although anyone who creates a **terminfo** entry for a printer should use names consistent with the names found in user documents for the printer. Application developers should allow a user to specify a character set by number (leaving it up to the user to examine the **csnm** string to determine the correct number), or by name, where the application examines the **csnm** string to determine the corresponding character set number.

These capabilities are likely to be used only with dot-matrix printers. If they are not available, the strings should not be defined. For printers that have manually changed print-wheels or font cartridges, the boolean **daisy** is set.

## Section 2-6: Dot-Matrix Graphics

Dot-matrix printers typically have the capability of reproducing “raster-graphics” images. Three new numeric capabilities and three new string capabilities can help a program draw raster-graphics images independent of the type of dot-matrix printer or the number of pins or dots the printer can handle at one time.

### Dot-Matrix Graphics

<b>npins</b>	Number of pins, <i>N</i> , in print-head
<b>spinv</b>	Spacing of pins vertically in pins per inch
<b>spinh</b>	Spacing of dots horizontally in dots per inch
<b>porder</b>	Matches software bits to print-head pins
<b>sbim</b>	Start printing bit image graphics, <i>B</i> bits wide
<b>rbim</b>	End printing bit image graphics

The **sbim** string is used with a single argument,  $B$ , the width of the image in dots.

The model of dot-matrix or raster-graphics that **terminfo** presents is similar to the technique used for most dot-matrix printers: each pass of the printer's print-head is assumed to produce a dot-matrix that is  $N$  dots high and  $B$  dots wide. This is typically a wide, squat, rectangle of dots. The height of this rectangle in dots will vary from one printer to the next; this is given in the **npins** numeric capability. The size of the rectangle in fractions of an inch will also vary; it can be deduced from the **spinv** and **spinh** numeric capabilities.

With these three values an application can divide a complete raster-graphics image into several horizontal strips, perhaps interpolating to account for different dot spacing vertically and horizontally.

The **sbim** and **rbim** strings are used to start and end a dot-matrix image, respectively.

The **sbim** string is used with a single argument that gives the width of the dot-matrix in dots. A sequence of "image-data bytes" are sent to the printer after the **sbim** string and before the **rbim** string. The number of bytes is an integral multiple of the width of the dot-matrix; the multiple and the form of each byte is determined by the **porder** string as described below.

The **porder** string is a comma separated list of pin numbers optionally followed by an numerical offset. The offset, if given, is separated from the list with a semicolon. The position of each pin number in the list corresponds to a bit in an 8-bit data byte. The pins are numbered consecutively from 1 to **npins**, with 1 being the top pin. Note that the term "pin" is used loosely here; "ink-jet" dot-matrix printers don't have pins, but can be considered to have an equivalent method of applying a single dot of ink to paper. The bit positions in **porder** are in groups of 8, with the first position in each group the most significant bit and the last position the least significant bit. An application produces 8-bit bytes in the order of the groups in **porder**.

An application computes the "image-data bytes" from the internal image, mapping vertical dot positions in each print-head pass into 8-bit bytes, using a 1 bit where ink should be applied and 0 where no ink should be applied. This can be reversed (0 bit for ink, 1 bit for no ink) by giving a negative pin number. If a position is skipped in **porder**, a 0 bit is used. If a position has a lower case 'x' instead of a pin number, a 1 bit is used in the skipped position. For consistency, a lower case 'o' can be used to represent a 0 filled, skipped bit. There must be a multiple of 8 bit positions used or skipped in **porder**; if not, 0 bits are used to fill the last byte in the least significant bits. The offset, if given, is added to each data byte; the offset can be negative.

Some examples may help clarify the use of the **porder** string. The AT&T 470, AT&T 475 and C.Itoh 8510 printers provide eight pins for graphics. The pins are identified top to bottom by the 8 bits in a byte, from least significant to most. The **porder** strings for these printers would be **8,7,6,5,4,3,2,1**. The AT&T 478 and AT&T 479 printers also provide eight pins for graphics. However, the pins are identified in the reverse order. The **porder**



**Section 2-7: Effect of  
Changing Printing  
Resolution**

strings for these printers would be **1,2,3,4,5,6,7,8**. The AT&T 5310, AT&T 5320, DEC LA100, and DEC LN03 printers provide six pins for graphics. The pins are identified top to bottom by the decimal values 1, 2, 4, 8, 16 and 32. These correspond to the low six bits in an 8-bit byte, although the decimal values are further offset by the value 63. The **porder** string for these printers would be **„6,5,4,3,2,1;63**, or alternately **o,o,6,5,4,3,2,1;63**.

If the control sequences to change the character pitch or the line pitch are used, the pin or dot spacing may change:

Dot-Matrix Graphics  
Changing the Character/Line Pitches

**cpi** Change character pitch  
**cpix** If set, **cpi** changes **spinh**

**lpi** Change line pitch  
**lpix** If set, **lpi** changes **spinv**

Programs that use **cpi** or **lpi** should recalculate the dot spacing:

Dot-Matrix Graphics  
Effects of Changing the Character/Line Pitches

Before	<i>After</i>
Using <b>cpi</b> with <b>cpix</b> clear:	
$\$bold\ spinh\ '\$$	$\$bold\ spinh\$$
Using <b>cpi</b> with <b>cpix</b> set:	
$\$bold\ spinh\ '\$$	$\$bold\ spinh = bold\ spinh\ ' \cdot bold\ orhi\ over\ \{ bold\ \{orhi\ '\}\$$
Using <b>lpi</b> with <b>lpix</b> clear:	
$\$bold\ spinv\ '\$$	$\$bold\ spinv\$$
Using <b>lpi</b> with <b>lpix</b> set:	
$\$bold\ spinv\ '\$$	$\$bold\ spinv = bold\ \{spinv\ '\} \cdot bold\ orhi\ over\ \{ bold\ \{orhi\ '\}\$$
Using <b>chr</b> :	
$\$bold\ spinh\ '\$$	$\$bold\ spinh\$$
Using <b>cvr</b> :	
$\$bold\ spinv\ '\$$	$\$bold\ spinv\$$

**orhi'** and **orhi** are the values of the horizontal resolution in steps per inch, before using **cpi** and after using **cpi**, respectively. Likewise, **orvi'** and **orvi** are the values of the vertical resolution in steps per inch, before using **lpi** and after using **lpi**, respectively. Thus, the changes in the dots per inch for dot-matrix graphics follow the changes in steps per inch for printer resolution.

**Section 2-8: Print Quality**

Many dot-matrix printers can alter the dot spacing of printed text to produce near “letter quality” printing or “draft quality” printing. Usually it is important to be able to choose one or the other because the rate of printing generally falls off as the quality improves. There are three new strings used to describe these capabilities.

Print Quality	
<b>snlq</b>	Set near-letter quality print
<b>srmq</b>	Set normal quality print
<b>sdrfq</b>	Set draft quality print

The capabilities are listed in decreasing levels of quality. If a printer doesn't have all three levels, one or two of the strings should be left blank as appropriate.

**Section 2-9: Printing Rate and Buffer Size**

Because there is no standard protocol that can be used to keep a program synchronized with a printer, and because modern printers can buffer data before printing it, a program generally cannot determine at any time what has been printed. Two new numeric capabilities can help a program estimate what has been printed.

Print Rate/Buffer Size	
<b>cps</b>	Nominal print rate in characters per second
<b>bufsz</b>	Buffer capacity in characters

**cps** is the nominal or average rate at which the printer prints characters; if this value is not given, the rate should be estimated at one-tenth the prevailing baud rate. **bufsz** is the maximum number of subsequent characters buffered before the guaranteed printing of an earlier character, assuming proper flow control has been used. If this value is not given it is assumed that the printer does not buffer characters, but prints them as they are received.

As an example, if a printer has a 1000-character buffer, then sending the letter “a” followed by 1000 additional characters is guaranteed to cause the letter “a” to print. If the same printer prints at the rate of 100 characters per second, then it should take 10 seconds to print all the characters in the buffer, less if the buffer is not full. By keeping track of the characters sent to a printer, and knowing the print rate and buffer size, a program can synchronize itself with the printer.

Note that most printer manufacturers advertise the maximum print rate, not the nominal print rate. A good way to get a value to put in for **cps** is to generate a few pages of text, count the number of printable characters, and then see how long it takes to print the text.

Applications that use these values should recognize the variability in the print rate. Straight text, in short lines, with no embedded control sequences will probably print at close to the advertised print rate and probably faster than the rate in **cps**. Graphics data with a lot of control sequences, or very long lines of text, will print at well below the advertised rate and below the rate in **cps**. If the application is using **cps** to decide how

long it should take a printer to print a block of text, the application should pad the estimate. If the application is using **cps** to decide how much text has already been printed, it should shrink the estimate. The application will thus err in favor of the user, who wants, above all, to see all the output in its correct place.

**FILES**

<b>/usr/share/lib/terminfo/?/*</b>	compiled terminal description database
<b>/usr/share/lib/.COREterm/?/*</b>	subset of compiled terminal description database
<b>/usr/share/lib/tabset/*</b>	tab settings for some terminals, in a format appropriate to be output to the terminal (escape sequences that set margins and tabs)

**SEE ALSO** **ls(1)**, **pg(1)**, **stty(1)**, **tput(1)**, **tty(1)**, **vi(1)**, **infocmp(1M)**, **tic(1M)**, **printf(3S)**, **curses(3X)**, **curses(3XC)**

**NOTES** The most effective way to prepare a terminal description is by imitating the description of a similar terminal in **terminfo** and to build up a description gradually, using partial descriptions with a screen oriented editor, such as **vi**, to check that they are correct. To easily test a new terminal description the environment variable **TERMINFO** can be set to the pathname of a directory containing the compiled description, and programs will look there rather than in **/usr/share/lib/terminfo**.

<b>NAME</b>	TIMEZONE – set default system time zone and locale
<b>SYNOPSIS</b>	<b>/etc/TIMEZONE</b> <b>/etc/default/init</b>
<b>DESCRIPTION</b>	This file sets the time zone environment variable <b>TZ</b> , and the locale-related environment variables <b>LANG</b> , <b>LC_COLLATE</b> , <b>LC_CTYPE</b> , <b>LC_MESSAGES</b> , <b>LC_MONETARY</b> , <b>LC_NUMERIC</b> , and <b>LC_TIME</b> . <b>/etc/TIMEZONE</b> is a symbolic link to <b>/etc/default/init</b> . The number of environments that can be set from <b>/etc/default/init</b> is limited to 20.
<b>SEE ALSO</b>	<b>init(1M)</b> , <b>ctime(3C)</b> , <b>environ(5)</b>

<b>NAME</b>	timezone – default timezone data base
<b>SYNOPSIS</b>	<b>/etc/timezone</b>
<b>DESCRIPTION</b>	<p>The timezone file contains information regarding the default timezone for each host in a domain. Alternatively, a single default line for the entire domain may be specified. Each entry has the format:</p> <p style="padding-left: 40px;"><i>Timezone-name official-host-or-domain-name</i></p> <p>Items are separated by any number of blanks and/or TAB characters. A '#' indicates the beginning of a comment; characters up to the end of the line are not interpreted by routines which search the file. The timezone is a pathname relative to the directory <b>/usr/share/lib/zoneinfo</b>.</p> <p>This file is not actually referenced by any system software; it is merely used as a source file to construct the NIS <b>timezone.byname</b> map. This map is read by the program <b>/usr/etc/install/sysIDtool</b> to initialize the timezone of the client system at installation time.</p> <p>The <b>timzone</b> file does not set the timezone environment variable <b>TZ</b>. See <b>TIMEZONE(4)</b> for information to set the <b>TZ</b> environment variable.</p>
<b>EXAMPLES</b>	<p>Here is a typical line from the <b>/etc/timezone</b> file:</p> <p><b>US/Eastern                    East.Sun.COM #Sun East Coast</b></p>
<b>FILES</b>	<b>/etc/timezone</b>
<b>SEE ALSO</b>	<b>TIMEZONE(4)</b>

<b>NAME</b>	tnf_kernel_probes – TNF kernel probes																				
<b>DESCRIPTION</b>	<p>The set of probes (trace instrumentation points) available in the standard kernel. The probes log trace data to a kernel trace buffer in Trace Normal Form (TNF). Kernel probes are controlled by <b>prex(1)</b>. A snapshot of the kernel trace buffer can be made using <b>tnfextract(1)</b> and examined using <b>tnfdump(1)</b>.</p> <p>Each probe has a <i>name</i> and is associated with a set of symbolic <i>keys</i>, or <i>categories</i>. These are used to select and control probes from <b>prex(1)</b>. A probe that is enabled for tracing generates a TNF record, called an <i>event record</i>. An event record contains two common members and may contain other probe-specific data members.</p>																				
<b>Common Members</b>	<table border="0"> <thead> <tr> <th style="text-align: left;">tnf type name</th> <th style="text-align: left;">member name</th> </tr> </thead> <tbody> <tr> <td><b>tnf_probe_event</b></td> <td><i>tag</i></td> </tr> <tr> <td><b>tnf_time_delta</b></td> <td><i>time_delta</i></td> </tr> </tbody> </table> <p><i>tag</i> encodes TNF references to two other records:</p> <table border="0"> <tr> <td style="padding-left: 2em;"><i>tag</i></td> <td>describes the layout of the event record</td> </tr> <tr> <td style="padding-left: 2em;"><i>schedule</i></td> <td>identifies the writing thread and also contains a 64-bit base time in nanoseconds.</td> </tr> </table> <p><i>time_delta</i> a 32-bit time offset from the base time; the sum of the two times is the actual time of the event.</p>	tnf type name	member name	<b>tnf_probe_event</b>	<i>tag</i>	<b>tnf_time_delta</b>	<i>time_delta</i>	<i>tag</i>	describes the layout of the event record	<i>schedule</i>	identifies the writing thread and also contains a 64-bit base time in nanoseconds.										
tnf type name	member name																				
<b>tnf_probe_event</b>	<i>tag</i>																				
<b>tnf_time_delta</b>	<i>time_delta</i>																				
<i>tag</i>	describes the layout of the event record																				
<i>schedule</i>	identifies the writing thread and also contains a 64-bit base time in nanoseconds.																				
<b>Threads</b>	<p><b>thread_create</b></p> <table border="0"> <thead> <tr> <th style="text-align: left;">tnf_kthread_id</th> <th style="text-align: left;"><i>tid</i></th> </tr> <tr> <th style="text-align: left;">tnf_pid</th> <th style="text-align: left;"><i>pid</i></th> </tr> <tr> <th style="text-align: left;">tnf_symbol</th> <th style="text-align: left;"><i>start_pc</i></th> </tr> </thead> </table> <p>Thread creation event.</p> <table border="0"> <tr> <td style="padding-left: 2em;"><i>tid</i></td> <td>the thread identifier for the new thread</td> </tr> <tr> <td style="padding-left: 2em;"><i>pid</i></td> <td>the process identifier for the new thread</td> </tr> <tr> <td style="padding-left: 2em;"><i>start_pc</i></td> <td>the kernel address of its start routine.</td> </tr> </table> <p><b>thread_state</b></p> <table border="0"> <thead> <tr> <th style="text-align: left;">tnf_kthread_id</th> <th style="text-align: left;"><i>tid</i></th> </tr> <tr> <th style="text-align: left;">tnf_microstate</th> <th style="text-align: left;"><i>state</i></th> </tr> </thead> </table> <p>Thread microstate transition events.</p> <table border="0"> <tr> <td style="padding-left: 2em;"><i>tid</i></td> <td>optional; if it is absent, the event is for the writing thread, otherwise the event is for the specified thread.</td> </tr> <tr> <td style="padding-left: 2em;"><i>state</i></td> <td>indicates the thread state: <ul style="list-style-type: none"> <li>• running in user mode</li> <li>• running in system mode</li> <li>• asleep waiting for a user-mode lock</li> <li>• asleep on a kernel object,</li> <li>• runnable (waiting for a cpu)</li> </ul> </td> </tr> </table>	tnf_kthread_id	<i>tid</i>	tnf_pid	<i>pid</i>	tnf_symbol	<i>start_pc</i>	<i>tid</i>	the thread identifier for the new thread	<i>pid</i>	the process identifier for the new thread	<i>start_pc</i>	the kernel address of its start routine.	tnf_kthread_id	<i>tid</i>	tnf_microstate	<i>state</i>	<i>tid</i>	optional; if it is absent, the event is for the writing thread, otherwise the event is for the specified thread.	<i>state</i>	indicates the thread state: <ul style="list-style-type: none"> <li>• running in user mode</li> <li>• running in system mode</li> <li>• asleep waiting for a user-mode lock</li> <li>• asleep on a kernel object,</li> <li>• runnable (waiting for a cpu)</li> </ul>
tnf_kthread_id	<i>tid</i>																				
tnf_pid	<i>pid</i>																				
tnf_symbol	<i>start_pc</i>																				
<i>tid</i>	the thread identifier for the new thread																				
<i>pid</i>	the process identifier for the new thread																				
<i>start_pc</i>	the kernel address of its start routine.																				
tnf_kthread_id	<i>tid</i>																				
tnf_microstate	<i>state</i>																				
<i>tid</i>	optional; if it is absent, the event is for the writing thread, otherwise the event is for the specified thread.																				
<i>state</i>	indicates the thread state: <ul style="list-style-type: none"> <li>• running in user mode</li> <li>• running in system mode</li> <li>• asleep waiting for a user-mode lock</li> <li>• asleep on a kernel object,</li> <li>• runnable (waiting for a cpu)</li> </ul>																				

- stopped.

The values of this member are defined in `<sys/msacct.h>`. Note that to reduce trace output, transitions between the *system* and *user* microstates that are induced by system calls are not traced. This information is implicit in the system call entry and exit events.

### thread\_exit

Thread termination event for writing thread. This probe has no data members other than the common members.

## Scheduling

### thread\_queue

<b>tnf_kthread_id</b>	<i>tid</i>
<b>tnf_cpuid</b>	<i>cpuid</i>
<b>tnf_long</b>	<i>priority</i>
<b>tnf_ulong</b>	<i>queue_length</i>

Thread scheduling events. These are triggered when a runnable thread is placed on a dispatch queue.

<i>cpuid</i>	specifies the cpu to which the queue is attached.
<i>priority</i>	the (global) dispatch priority of the thread.
<i>queue_length</i>	the current length of the cpu's dispatch queue.

## Blocking

### thread\_block

<b>tnf_opaque</b>	<i>reason</i>
<b>tnf_symbols</b>	<i>stack</i>

Thread blockage event. This probe captures a partial stack backtrace when the current thread blocks.

<i>reason</i>	the address of the object on which the thread is blocking.
<i>symbols</i>	references a TNF array of kernel addresses representing the PCs on the stack at the time the thread blocks.

## System Calls

### syscall\_start

<b>tnf_sysnum</b>	<i>sysnum</i>
-------------------	---------------

System call entry event.

<i>sysnum</i>	the system call number. The writing thread implicitly enters the <i>system</i> microstate with this event.
---------------	--

### syscall\_end

<b>tnf_long</b>	<i>rval1</i>
<b>tnf_long</b>	<i>rval2</i>
<b>tnf_long</b>	<i>errno</i>

System call exit event.

<i>rval1</i>	the two return values of the system call
<i>rval2</i>	

**Page Faults**

*errno* the error return.

The writing thread implicitly enters the *user* microstate with this event.

**address\_fault**

**tnf\_opaque** *address*  
**tnf\_fault\_type** *fault\_type*  
**tnf\_seg\_access** *access*

Address-space fault event.

*address* gives the faulting virtual address.

*fault\_type* gives the fault type: invalid page, protection fault, software requested locking or unlocking.

*access* gives the desired access protection: read, write, execute or create.

The values for these two members are defined in `<vm/seg_enum.h>`.

**major\_fault**

**tnf\_opaque** *vnode*  
**tnf\_offset** *offset*

Major page fault event. The faulting page is mapped to the file given by the *vnode* member, at the given *offset* into the file. (The faulting virtual address is in the most recent **address\_fault** event for the writing thread.)

**anon\_private**

**tnf\_opaque** *address*

Copy-on-write page fault event.

*address* the virtual address at which the new page is mapped.

**anon\_zero**

**tnf\_opaque** *address*

Zero-fill page fault event.

*address* the virtual address at which the new page is mapped.

**page\_unmap**

**tnf\_opaque** *vnode*  
**tnf\_offset** *offset*

Page unmapping event. This probe marks the unmapping of a file system page from the system.

*vnode* and *offset*

identify the file and offset of the page being unmapped.

**Pageins and Pageouts**



**pagein**

**tnf\_opaque** *vnode*  
**tnf\_offset** *offset*  
**tnf\_size** *size*

Pagein start event. This event signals the initiation of pagein I/O.

*vnode* and *offset*

identify the file and offset to be paged in.

*size* specifies the number of bytes to be paged in.

**pageout**

**tnf\_opaque** *vnode*  
**tnf\_ulong** *pages\_pageout*  
**tnf\_ulong** *pages\_freed*  
**tnf\_ulong** *pages\_reclaimed*

Pageout completion event. This event signals the completion of pageout I/O.

*vnode* identifies the file of the pageout request.

*pages\_pageout* the number of pages written out.

*pages\_freed* the number of pages freed after being written out.

*pages\_reclaimed* the number of pages reclaimed after being written out.

**Page Daemon (Page Stealer)****pageout\_scan\_start**

**tnf\_ulong** *pages\_free*  
**tnf\_ulong** *pages\_needed*

Page daemon scan start event. This event signals the beginning of one iteration of the page daemon.

*pages\_free* the number of free pages in the system.

*pages\_needed* the number of pages desired free.

**pageout\_scan\_end**

**tnf\_ulong** *pages\_free*  
**tnf\_ulong** *pages\_scanned*

Page daemon scan end event. This event signals the end of one iteration of the page daemon.

*pages\_free* the number of free pages in the system.

*pages\_scanned* the number of pages examined by the page daemon. (Potentially more pages will be freed when any queued pageout requests complete.)

**Swapper****swapout\_process**

**tnf\_pid** *pid*  
**tnf\_ulong** *page\_count*

Address space swapout event. This event marks the swapping out of a process address space.

*pid* identifies the process.  
*page\_count* reports the number of pages either freed or queued for pageout.

**swapout\_lwp**

**tnf\_pid** *pid*  
**tnf\_lwpid** *lwpid*  
**tnf\_kthread\_id** *tid*  
**tnf\_ulong** *page\_count*

Light-weight process swapout event. This event marks the swapping out of an LWP and its stack.

*pid* the LWP's process identifier  
*lwpid* the LWP identifier  
*tid member* the LWP's kernel thread identifier.  
*page\_count* the number of pages swapped out.

**swapi\_lwp**

**tnf\_pid** *pid*  
**tnf\_lwpid** *lwpid*  
**tnf\_kthread\_id** *tid*  
**tnf\_ulong** *page\_count*

Light-weight process swapi event. This event marks the swapping in of an LWP and its stack.

*pid* the LWP's process identifier  
*lwpid* the LWP identifier  
*tid* the LWP's kernel thread identifier.  
*page\_count* the number of pages swapped in.

**Local I/O****strategy**

**tnf\_device** *device*  
**tnf\_diskaddr** *block*  
**tnf\_size** *size*  
**tnf\_opaque** *buf*  
**tnf\_bioflags** *flags*

Block I/O strategy event. This event marks a call to the **strategy**(9E) routine of a block device driver.

*device* contains the major and minor numbers of the device.  
*block* the logical block number to be accessed on the device.  
*size* the size of the I/O request.  
*buf* the kernel address of the **buf**(9S) structure associated with the transfer.  
*flags* the **buf**(9S) flags associated with the transfer.

**biodone**

**tnf\_device** *device*  
**tnf\_diskaddr** *block*  
**tnf\_opaque** *buf*

Buffered I/O completion event. This event marks calls to the **biodone**(9F) routine.

*device* contains the major and minor numbers of the device.

*block* the logical block number accessed on the device.

*buf* the kernel address of the **buf**(9S) structure associated with the transfer.

**physio\_start**

**tnf\_device** *device*

**tnf\_offset** *offset*

**tnf\_size** *size*

**tnf\_bioflags** *rw*

Raw I/O start event. This event marks entry into the **physio**(9F) routine which performs unbuffered I/O.

*device* contains the major and minor numbers of the device of the transfer.

*offset* the logical offset on the device for the transfer.

*size* the number of bytes to be transferred.

*rw* the direction of the transfer: read or write (see **buf**(9S)).

**physio\_end**

**tnf\_device** *device*

Raw I/O end event. This event marks exit from the **physio**(9F) routine.

*device* the major and minor numbers of the device of the transfer.

**SEE ALSO**

**prex**(1), **tnfdump**(1), **tnfextract**(1), **libtnfctl**(3X), **TNF\_PROBE**(3X), **tracing**(3X), **strategy**(9E), **biodone**(9F), **physio**(9F), **buf**(9S)

<b>NAME</b>	ts_dptbl – time-sharing dispatcher parameter table
<b>DESCRIPTION</b>	<p>The process scheduler (or dispatcher) is the portion of the kernel that controls allocation of the CPU to processes. The scheduler supports the notion of scheduling classes where each class defines a scheduling policy, used to schedule processes within that class. Associated with each scheduling class is a set of priority queues on which ready to run processes are linked. These priority queues are mapped by the system configuration into a set of global scheduling priorities which are available to processes within the class. (The dispatcher always selects for execution the process with the highest global scheduling priority in the system.) The priority queues associated with a given class are viewed by that class as a contiguous set of priority levels numbered from 0 (lowest priority) to <i>n</i> (highest priority—a configuration-dependent value). The set of global scheduling priorities that the queues for a given class are mapped into might not start at zero and might not be contiguous (depending on the configuration).</p> <p>Processes in the time-sharing class which are running in user mode (or in kernel mode before going to sleep) are scheduled according to the parameters in a time-sharing dispatcher parameter table (<b>ts_dptbl</b>). Processes in the inter-active scheduling class are also scheduled according to the parameters in the time-sharing dispatcher parameter table. (Time-sharing processes and inter-active processes running in kernel mode after sleeping are run within a special range of priorities reserved for such processes and are not affected by the parameters in the <b>ts_dptbl</b> until they return to user mode.) The <b>ts_dptbl</b> consists of an array (<b>config_ts_dptbl[]</b>) of parameter structures (<b>struct tsdpent_t</b>), one for each of the <i>n</i> priority levels used by time-sharing processes and inter-active processes in user mode. The structures are accessed via a pointer, (<b>ts_dptbl</b>), to the array. The properties of a given priority level <i>i</i> are specified by the <i>i</i>th parameter structure in this array (<b>ts_dptbl[ i ]</b>).</p> <p>A parameter structure consists of the following members. These are also described in the <b>/usr/include/sys/ts.h</b> header.</p> <p><b>ts_globpri</b> The global scheduling priority associated with this priority level. The mapping between time-sharing priority levels and global scheduling priorities is determined at boot time by the system configuration. <b>ts_globpri</b> is the only member of the <b>ts_dptbl</b> which cannot be changed with <b>dispadm(1M)</b>.</p> <p><b>ts_quantum</b> The length of the time quantum allocated to processes at this level in ticks (Hz).</p> <p><b>ts_tqexp</b> Priority level of the new queue on which to place a process running at the current level if it exceeds its time quantum. Normally this field links to a lower priority time-sharing level that has a larger quantum.</p>

**ts\_slpret**

Priority level of the new queue on which to place a process, that was previously in user mode at this level, when it returns to user mode after sleeping. Normally this field links to a higher priority level that has a smaller quantum.

**ts\_maxwait**

A per process counter, **ts\_dispwait** is initialized to zero each time a time-sharing or inter-active process is placed back on the dispatcher queue after its time quantum has expired or when it is awakened (**ts\_dispwait** is not reset to zero when a process is preempted by a higher priority process). This counter is incremented once per second for each process on the dispatcher queue. If a process's **ts\_dispwait** value exceeds the **ts\_maxwait** value for its level, the process's priority is changed to that indicated by **ts\_lwait**. The purpose of this field is to prevent starvation.

**ts\_lwait** Move a process to this new priority level if **ts\_dispwait** is greater than **ts\_maxwait**.

An administrator can affect the behavior of the time-sharing portion of the scheduler by reconfiguring the **ts\_dptbl**. Since processes in the time-sharing and inter-active scheduling classes share the same dispatch parameter table (**ts\_dptbl**), changes to this table will affect both scheduling classes. There are two methods available for doing this: reconfigure with a loadable module at boot-time or by using **dispadmin(1M)** at run-time.

**TS\_DPTBL  
LOADABLE  
MODULE**

The **ts\_dptbl** can be reconfigured with a loadable module which contains a new time sharing dispatch table. The module containing the dispatch table is separate from the TS loadable module which contains the rest of the time-sharing and inter-active software. This is the only method that can be used to change the number of time-sharing priority levels or the set of global scheduling priorities used by the time-sharing and inter-active classes. The relevant procedure and source code is described in the **REPLACING THE TS\_DPTBL LOADABLE MODULE** section.

**DISPADMIN  
CONFIGURATION  
FILE**

With the exception of **ts\_globpri** all of the members of the **ts\_dptbl** can be examined and modified on a running system using the **dispadmin(1M)** command. Invoking **dispadmin** for the time-sharing or inter-active class allows the administrator to retrieve the current **ts\_dptbl** configuration from the kernel's in-core table, or overwrite the in-core table with values from a configuration file. The configuration file used for input to **dispadmin** must conform to the specific format described below.

Blank lines are ignored and any part of a line to the right of a # symbol is treated as a comment. The first non-blank, non-comment line must indicate the resolution to be used for interpreting the **ts\_quantum** time quantum values. The resolution is specified as

**RES=res**

where *res* is a positive integer between 1 and 1,000,000,000 inclusive and the resolution used is the reciprocal of *res* in seconds (for example, **RES=1000** specifies millisecond resolution). Although very fine (nanosecond) resolution may be specified, the time quantum lengths are rounded up to the next integral multiple of the system clock's resolution.

The remaining lines in the file are used to specify the parameter values for each of the time-sharing priority levels. The first line specifies the parameters for time-sharing level 0, the second line specifies the parameters for time-sharing level 1, etc. There must be exactly one line for each configured time-sharing priority level.

### EXAMPLES

The following excerpt from a **dispadmin** configuration file illustrates the format. Note that for each line specifying a set of parameters there is a comment indicating the corresponding priority level. These level numbers indicate priority within the time-sharing and inter-active classes, and the mapping between these time-sharing priorities and the corresponding global scheduling priorities is determined by the configuration specified in the **ts** master file. The level numbers are strictly for the convenience of the administrator reading the file and, as with any comment, they are ignored by **dispadmin**. **dispadmin** assumes that the lines in the file are ordered by consecutive, increasing priority level (from 0 to the maximum configured time-sharing priority). The level numbers in the comments should normally agree with this ordering; if for some reason they don't, however, **dispadmin** is unaffected.

```
# Time-Sharing Dispatcher Configuration File RES=1000
# ts_quantum  ts_tqexp  ts_slpret  ts_maxwait  ts_lwait  PRIORITY
#                                                    LEVEL
  500          0        10          5          10        #  0
  500          0        11          5          11        #  1
  500          1        12          5          12        #  2
  500          1        13          5          13        #  3
  500          2        14          5          14        #  4
  500          2        15          5          15        #  5
  450          3        16          5          16        #  6
  450          3        17          5          17        #  7
  .            .        .            .            .        . .
  .            .        .            .            .        . .
  .            .        .            .            .        . .
  50          48        59          5          59        # 58
  50          49        59          5          59        # 59
```

### REPLACING THE TS\_DPTBL LOADABLE MODULE

In order to change the size of the time sharing dispatch table, the loadable module which contains the dispatch table information will have to be built. It is recommended that you save the existing module before using the following procedure.

1. Place the dispatch table code shown below in a file called **ts\_dptbl.c**. An example of this file follows.
2. Compile the code using the given compilation and link lines supplied.  

```
cc -c -O -D_KERNEL ts_dptbl.c
ld -r -o TS_DPTBL ts_dptbl.o
```
3. Copy the current dispatch table in **/kernel/sched** to **TS\_DPTBL.bak**.
4. Replace the current **TS\_DPTBL** in **/kernel/sched**.

5. You will have to make changes in the `/etc/system` file to reflect the changes to the sizes of the tables. See `system(4)`. The two variables affected are `ts_maxupri` and `ts_maxkmdpri`. The syntax for setting these is as follows:

```
set TS:ts_maxupri=(value for max time-sharing user priority)
set TS:ts_maxkmdpri=(number of kernel mode priorities - 1)
```

6. Reboot the system to use the new dispatch table.

**NOTE: Great care should be used in replacing the dispatch table using this method. If you do not get it right, panics may result, thus making the system unusable.**

The following is an example of a `ts_dptbl.c` file used for building the new `ts_dptbl`.

```
/* BEGIN ts_dptbl.c */

#include <sys/proc.h>
#include <sys/priocntl.h>
#include <sys/class.h>
#include <sys/disp.h>
#include <sys/ts.h>
#include <sys/rtpriocntl.h>

/*
 * This is the loadable module wrapper.
 */
#include <sys/modctl.h>

extern struct mod_ops mod_miscops;

/*
 * Module linkage information for the kernel.
 */
static struct modlmisc modlmisc = {
    &mod_miscops, "Time sharing dispatch table"
};

static struct modlinkage modlinkage = {
    MODREV_1, &modlmisc, 0
};

_init()
{
    return (mod_install(&modlinkage));
}
```

```

_info(modinfo)
    struct modinfo *modinfo;
{
    return (mod_info(&modlinkage, modinfo));
}

/*
 * array of global priorities used by ts procs sleeping or
 * running in kernel mode after sleep. Must have at least
 * 40 values.
 */

pri_t config_ts_kmdpris[] = {
    60,61,62,63,64,65,66,67,68,69,
    70,71,72,73,74,75,76,77,78,79,
    80,81,82,83,84,85,86,87,88,89,
    90,91,92,93,94,95,96,97,98,99,
};

tsdpent_t          config_ts_dptbl[] = {
/* glbpri   qntm   tqexp   slprt   mxwt   lwt */
    0,    100,    0,    10,    5,    10,
    1,    100,    0,    11,    5,    11,
    2,    100,    1,    12,    5,    12,
    3,    100,    1,    13,    5,    13,
    4,    100,    2,    14,    5,    14,
    5,    100,    2,    15,    5,    15,
    6,    100,    3,    16,    5,    16,
    7,    100,    3,    17,    5,    17,
    8,    100,    4,    18,    5,    18,
    9,    100,    4,    19,    5,    19,
    10,   80,    5,    20,    5,    20,
    11,   80,    5,    21,    5,    21,
    12,   80,    6,    22,    5,    22,
    13,   80,    6,    23,    5,    23,
    14,   80,    7,    24,    5,    24,
    15,   80,    7,    25,    5,    25,
    16,   80,    8,    26,    5,    26,
    17,   80,    8,    27,    5,    27,
    18,   80,    9,    28,    5,    28,
    19,   80,    9,    29,    5,    29,
    20,   60,   10,   30,    5,    30,
    21,   60,   11,   31,    5,    31,

```



```

22,    60,    12,    32,    5,    32,
23,    60,    13,    33,    5,    33,
24,    60,    14,    34,    5,    34,
25,    60,    15,    35,    5,    35,
26,    60,    16,    36,    5,    36,
27,    60,    17,    37,    5,    37,
28,    60,    18,    38,    5,    38,
29,    60,    19,    39,    5,    39,
30,    40,    20,    40,    5,    40,
31,    40,    21,    41,    5,    41,
32,    40,    22,    42,    5,    42,
33,    40,    23,    43,    5,    43,
34,    40,    24,    44,    5,    44,
35,    40,    25,    45,    5,    45,
36,    40,    26,    46,    5,    46,
37,    40,    27,    47,    5,    47,
38,    40,    28,    48,    5,    48,
39,    40,    29,    49,    5,    49,
40,    20,    30,    50,    5,    50,
41,    20,    31,    50,    5,    50,
42,    20,    32,    51,    5,    51,
43,    20,    33,    51,    5,    51,
44,    20,    34,    52,    5,    52,
45,    20,    35,    52,    5,    52,
46,    20,    36,    53,    5,    53,
47,    20,    37,    53,    5,    53,
48,    20,    38,    54,    5,    54,
49,    20,    39,    54,    5,    54,
50,    10,    40,    55,    5,    55,
51,    10,    41,    55,    5,    55,
52,    10,    42,    56,    5,    56,
53,    10,    43,    56,    5,    56,
54,    10,    44,    57,    5,    57,
55,    10,    45,    57,    5,    57,
56,    10,    46,    58,    5,    58,
57,    10,    47,    58,    5,    58,
58,    10,    48,    59,    5,    59,
59,    10,    49,    59,    5,    59,

```

```
};
```

```
short config_ts_maxumdpr = sizeof (config_ts_dptbl)/16 - 1;
```

```
/*
```

```
* Return the address of config_ts_dptbl
```

```
*/
```

```

tsdpent_t *
ts_getdptbl()
{
    return (config_ts_dptbl);
}

/*
 * Return the address of config_ts_kmdpris
 */
int *
ts_getkmdpris()
{
    return (config_ts_kmdpris);
}

/*
 * Return the address of ts_maxumdpr
 */
short
ts_getmaxumdpr()
{
    return (config_ts_maxumdpr);
}

/* END ts_dptbl.c */

```

**FILES** <sys/ts.h>

**SEE ALSO** **priocntl(1)**, **dispadm(1M)**, **priocntl(2)**, **system(4)**

*System Administration Guide*  
*System Interface Guide*

**NOTES** **dispadm** does some limited sanity checking on the values supplied in the configuration file. The sanity checking is intended to ensure that the new **ts\_dptbl** values do not cause the system to panic. The sanity checking does not attempt to analyze the effect that the new values will have on the performance of the system. Unusual **ts\_dptbl** configurations may have a dramatic negative impact on the performance of the system.

No sanity checking is done on the **ts\_dptbl** values specified in the **TS\_DPTBL** loadable module. Specifying an inconsistent or nonsensical **ts\_dptbl** configuration through the **TS\_DPTBL** loadable module could cause serious performance problems and/or cause the system to panic.

<b>NAME</b>	ttydefs – file contains terminal line settings information for ttymon
<b>DESCRIPTION</b>	<p><b>/etc/ttydefs</b> is an administrative file that contains records divided into fields by colons (":"). This information used by <b>ttymon</b> to set up the speed and terminal settings for a TTY port.</p> <p>The <b>ttydefs</b> file contains the following fields:</p> <p><i>tylabel</i>            The string <b>ttymon</b> tries to match against the TTY port's <i>tylabel</i> field in the port monitor administrative file. It often describes the speed at which the terminal is supposed to run, for example, <b>1200</b>.</p> <p><i>initial-flags</i>      Contains the initial <b>termio(7I)</b> settings to which the terminal is to be set. For example, the system administrator will be able to specify what the default erase and kill characters will be. <i>initial-flags</i> must be specified in the syntax recognized by the <b>stty</b> command.</p> <p><i>final-flags</i>        <i>final-flags</i> must be specified in the same format as <i>initial-flags</i>. <b>ttymon</b> sets these final settings after a connection request has been made and immediately prior to invoking a port's service.</p> <p><i>autobaud</i>           If the autobaud field contains the character 'A,' autobaud will be enabled. Otherwise, autobaud will be disabled. <b>ttymon</b> determines what line speed to set the TTY port to by analyzing the carriage returns entered. If autobaud has been disabled, the hunt sequence is used for baud rate determination.</p> <p><i>nextlabel</i>          If the user indicates that the current terminal setting is not appropriate by sending a BREAK, <b>ttymon</b> searches for a <b>ttydefs</b> entry whose <i>tylabel</i> field matches the <i>nextlabel</i> field. If a match is found, <b>ttymon</b> uses that field as its <i>tylabel</i> field. A series of speeds is often linked together in this way into a closed set called a hunt sequence. For example, <b>4800</b> may be linked to <b>1200</b>, which in turn is linked to <b>2400</b>, which is finally linked to <b>4800</b>.</p>
<b>SEE ALSO</b>	<p><b>sttydefs(1M)</b>, <b>ttymon(1M)</b>, <b>termio(7I)</b></p> <p><i>System Administration Guide</i></p>

<b>NAME</b>	<b>ttsrch</b> – directory search list for ttyname
<b>DESCRIPTION</b>	<p><b>ttsrch</b> is an optional file that is used by the <b>ttyname</b> library routine. This file contains the names of directories in <b>/dev</b> that contain terminal and terminal-related device files. The purpose of this file is to improve the performance of <b>ttyname</b> by indicating which subdirectories in <b>/dev</b> contain terminal-related device files and should be searched first. These subdirectory names must appear on separate lines and must begin with <b>/dev</b>. Those path names that do not begin with <b>/dev</b> will be ignored and a warning will be sent to the console. Blank lines (lines containing only white space) and lines beginning with the comment character "#" will be ignored. For each file listed (except for the special entry <b>/dev</b>), <b>ttyname</b> will recursively search through subdirectories looking for a match. If <b>/dev</b> appears in the <b>ttsrch</b> file, the <b>/dev</b> directory itself will be searched but there will not be a recursive search through its subdirectories.</p> <p>When <b>ttyname</b> searches through the device files, it tries to find a file whose major/minor device number, file system identifier, and inode number match that of the file descriptor it was given as an argument. If a match is not found, it will settle for a match of just major/minor device and file system identifier, if one can be found. However, if the file descriptor is associated with a cloned device, this algorithm does not work efficiently because the inode number of the device file associated with a clonable device will never match the inode number of the file descriptor that was returned by the open of that clonable device. To help with these situations, entries can be put into the <b>/etc/ttsrch</b> file to improve performance when cloned devices are used as terminals on a system (for example, for remote login). However, this is only useful if the minor devices related to a cloned device are put into a subdirectory. (It is important to note that device files need not exist for cloned devices and if that is the case, <b>ttyname</b> will eventually fail.) An optional second field is used in the <b>/etc/ttsrch</b> file to indicate the matching criteria. This field is separated by white space (any combination of blanks or tabs). The letter <b>M</b> means major/minor device number, <b>F</b> means file system identifier, and <b>I</b> means inode number. If this field is not specified for an entry, the default is <b>MFI</b> which means try to match on all three. For cloned devices the field should be <b>MF</b>, which indicates that it is not necessary to match on the inode number.</p> <p>Without the <b>/etc/ttsrch</b> file, <b>ttyname</b> will search the <b>/dev</b> directory by first looking in the directories <b>/dev/term</b>, <b>/dev/pts</b>, and <b>/dev/xt</b>. If a system has terminal devices installed in directories other than these, it may help performance if the <b>ttsrch</b> file is created and contains that list of directories.</p>
<b>EXAMPLES</b>	<p>A sample <b>/etc/ttsrch</b> file follows:</p> <pre> /dev/term      MFI /dev/pts       MFI /dev/xt        MFI /dev/slan      MF </pre>

This file tells **ttyname** that it should first search through those directories listed and that when searching through the **/dev/slans** directory, if a file is encountered whose major/minor devices and file system identifier match that of the file descriptor argument to **ttyname**, this device name should be considered a match.

**FILES** /etc/ttypsrch

**SEE ALSO** ttyname(3C)

**NAME** ufsdump, dumpdates – incremental dump format

**SYNOPSIS** `#include <sys/types.h>`  
`#include <sys/inode.h>`  
`#include <protocols/dumprestore.h>`  
`/etc/dumpdates`

**DESCRIPTION** Tapes used by `ufsdump(1M)` and `ufsrestore(1M)` contain:

- a header record
- two groups of bit map records
- a group of records describing directories
- a group of records describing files

The format of the header record and of the first record of each description as given in the include file `<protocols/dumprestore.h>` is:

```
#define TP_BSIZE          1024
#define NTREC             10
#define HIGHDENSITYTREC  32
#define CARTRIDGETREC    63
#define TP_NINDIR        (TP_BSIZE/2)
#define TP_NINOS         (TP_NINDIR / sizeop (long))
#define LBLSIZE          16
#define NAMELEN          64
#define NFS_MAGIC        (int) 60012
#define CHECKSUM         (int) 84446
union u_data {
    char s_addr[TP_NINDIR];
    long s_inos[TP_NINOS];
union u_spcl {
    char dummy[TP_BSIZE];
    struct s_spcl {
        long          c_type;
        time_t        c_date;
        time_t        c_ddate;
        long          c_volume;
        daddr_t       c_tapea;
        ino_t         c_inumber;
        long          c_magic;
        long          c_checksum;
        struct dinode c_dinode;
        long          c_count;
        union         u_data c_data;
        char          c_label[LBLSIZE];
```

```

        long          c_level;
        char          c_filesys[NAMELEN];
        char          c_dev[NAMELEN];
        char          c_host[NAMELEN];
        long          c_flags;
        long          c_firstrec;
        long          c_spare[32];
    } s_spcl;
} u_spcl;
#define spcl u_spcl.s_spcl
#define c_addr c_data.s_addr
#define c_inos cdata.s_inos
#define TS_TAPE          1
#define TS_INODE        2
#define TS_ADDR         4
#define TS_BITS         3
#define TS_CLRI         6
#define TS_END          5
#define TS_EOM          7
#define DR_NEWHEADER    1
#define DR_INODEINFO    2
#define DR_REDUMP       4
#define DR_TRUELIC      8
#define DUMPOUTFMT      "%-24s %c %s"
#define DUMPINFMT       "%24s %c %[\n]\n"

```

The constants are described as follows:

**TP\_BSIZE**        Size of file blocks on the dump tapes. Note that **TP\_BSIZE** must be a multiple of **DEV\_BSIZE**.

**NTREC**            Default number of **TP\_BSIZE** byte records in a physical tape block, changeable by the **b** option to **ufsdump**(1M).

**HIGHDENSITYNTREC**  
                   Default number of **TP\_BSIZE** byte records in a physical tape block on 6250 BPI or higher density tapes.

**CARTRIDGETREC**  
                   Default number of **TP\_BSIZE** records in a physical tape block on cartridge tapes.

**TP\_NINDIR**        Number of indirect pointers in a **TS\_INODE** or **TS\_ADDR** record. It must be a power of 2.

**TP\_NINOS**        The maximum number of volumes on a tape. Used for tape labeling in **hsmdump** and **hsmrestore** (available with Online:Backup 2.0 optional software package SUNWhsm).

- LBSIZE**           The maximum size of a volume label. Used for tape labeling in **hsmdump** and **hsmrestore** (available with Online:Backup 2.0 optional software package SUNWhsm).
- NAMELEN**        The maximum size of a host's name.
- NFS\_MAGIC**      All header records have this number in **c\_magic**.
- CHECKSUM**       Header records checksum to this value.

The **TS\_** entries are used in the **c\_type** field to indicate what sort of header this is. The types and their meanings are as follows:

- TS\_TAPE**        Tape volume label.
- TS\_INODE**      A file or directory follows. The **c\_dinode** field is a copy of the disk inode and contains bits telling what sort of file this is.
- TS\_ADDR**        A subrecord of a file description. See **s\_addrs** below.
- TS\_BITS**        A bit map follows. This bit map has a one bit for each inode that was dumped.
- TS\_CLRI**        A bit map follows. This bit map contains a zero bit for all inodes that were empty on the file system when dumped.
- TS\_END**         End of tape record.
- TS\_EOM**         floppy EOM — restore compat with old dump

The flags are described as follows:

- DR\_NEWHEADER**    New format tape header.
- DR\_INFODEINFO**   Header contains starting inode info.
- DR\_REDUMP**      Dump contains recopies of active files.
- DR\_TRUEINC**     Dump is a "true incremental".
- DUMPOUTFMT**    Name, incon, and ctime (date) for printf.
- DUMPINFMT**     Inverse for scanf.

The fields of the header structure are as follows:

- s\_addrs**        An array of bytes describing the blocks of the dumped file. A byte is zero if the block associated with that byte was not present on the file system; otherwise, the byte is non-zero. If the block was not present on the file system, no block was dumped; the block will be stored as a hole in the file. If there is not sufficient space in this record to describe all the blocks in a file, **TS\_ADDR** records will be scattered through the file, each one picking up where the last left off
- s\_inos**         The starting inodes on tape.
- c\_type**         The type of the record.
- c\_date**         The date of the previous dump.



<b>c_ddate</b>	The date of this dump.
<b>c_volume</b>	The current volume number of the dump.
<b>c_tapea</b>	The logical block of this record.
<b>c_inumber</b>	The number of the inode being dumped if this is of type <b>TS_INODE</b> .
<b>c_magic</b>	This contains the value <b>MAGIC</b> above, truncated as needed.
<b>c_checksum</b>	This contains whatever value is needed to make the record sum to <b>CHECKSUM</b> .
<b>c_dinode</b>	This is a copy of the inode as it appears on the file system.
<b>c_count</b>	The count of bytes in <b>s_addrs</b> .
<b>u_data c_data</b>	The union of either <b>u_data c_data</b> The union of either <b>s_addrs</b> or <b>s_inos</b> .
<b>c_label</b>	Label for this dump.
<b>c_level</b>	Level of this dump.
<b>c_filesys</b>	Name of dumped file system.
<b>c_dev</b>	Name of dumped service.
<b>c_host</b>	Name of dumped host.
<b>c_flags</b>	Additional information.
<b>c_firstrec</b>	First record on volume.
<b>c_spare</b>	Reserved for future uses.

Each volume except the last ends with a tapemark (read as an end of file). The last volume ends with a **TS\_END** record and then the tapemark.

The dump history is kept in the file **/etc/dumpdates**. It is an ASCII file with three fields separated by white space:

- The name of the device on which the dumped file system resides.
- The level number of the dump tape; see **ufsdump(1M)**.
- The date of the incremental dump in the format generated by **ctime(3C)**.

**DUMPOUTFMT** is the format to use when using **printf(3S)** to write an entry to **/etc/dumpdates**; **DUMPINFMT** is the format to use when using **scanf(3S)** to read an entry from **/etc/dumpdates**.

## ATTRIBUTES

See **attributes(5)** for a description of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Stability Level	Unstable

## SEE ALSO

**ufsdump(1M)**, **ufsrestore(1M)**, **ctime(3C)**, **printf(3S)**, **scanf(3S)**, **attributes(5)**, **types(5)**

<b>NAME</b>	unistd – header for symbolic constants																																
<b>SYNOPSIS</b>	<b>#include</b> <unistd.h>																																
<b>DESCRIPTION</b>	<p>The &lt;unistd.h&gt; header defines the symbolic constants and structures which are not already defined or declared in some other header. The contents of this header are shown below.</p> <p>The following symbolic constants are defined for the <b>access(2)</b> function:</p> <table border="0"> <tr> <td><b>R_OK</b></td> <td>Test for read permission.</td> </tr> <tr> <td><b>W_OK</b></td> <td>Test for write permission.</td> </tr> <tr> <td><b>X_OK</b></td> <td>Test for execute (search) permission.</td> </tr> <tr> <td><b>F_OK</b></td> <td>Test for existence of file.</td> </tr> </table> <p>The constants <b>F_OK</b>, <b>R_OK</b>, <b>W_OK</b>, and <b>X_OK</b>, and the expressions <b>R_OK   W_OK</b>, <b>R_OK   X_OK</b>, and <b>R_OK   W_OK   X_OK</b> all have distinct values.</p> <p>The following constant is declared:</p> <table border="0"> <tr> <td><b>NULL</b></td> <td>Null pointer</td> </tr> </table> <p>The following symbolic constants are defined for the <b>lockf(3C)</b> function:</p> <table border="0"> <tr> <td><b>F_ULOCK</b></td> <td>Unlock a previously locked region.</td> </tr> <tr> <td><b>F_LOCK</b></td> <td>Lock a region for exclusive use.</td> </tr> <tr> <td><b>F_TLOCK</b></td> <td>Test and lock a region for exclusive use.</td> </tr> <tr> <td><b>F_TEST</b></td> <td>Test a region for other processes locks.</td> </tr> </table> <p>The following symbolic constants are defined for the <b>lseek(2)</b> and <b>fcntl(2)</b> functions (they have distinct values):</p> <table border="0"> <tr> <td><b>SEEK_SET</b></td> <td>Set file offset to <i>offset</i>.</td> </tr> <tr> <td><b>SEEK_CUR</b></td> <td>Set file offset to current plus <i>offset</i>.</td> </tr> <tr> <td><b>SEEK_END</b></td> <td>Set file offset to EOF plus <i>offset</i>.</td> </tr> </table> <p>The following symbolic constants are defined (with fixed values):</p> <table border="0"> <tr> <td><b>_POSIX_VERSION</b></td> <td>Integer value indicating version of the POSIX standard (see <b>standards(5)</b>).</td> </tr> <tr> <td><b>_XOPEN_VERSION</b></td> <td>integer value indicating version of the XPG to which system conforms.</td> </tr> </table> <p>The following symbolic constants are defined to indicate that the option is present:</p> <table border="0"> <tr> <td><b>_POSIX_JOB_CONTROL</b></td> <td>Implementation supports job control.</td> </tr> <tr> <td><b>_POSIX_SAVED_IDS</b></td> <td>The exec functions (see <b>exec(2)</b>) save the effective user and group.</td> </tr> </table>	<b>R_OK</b>	Test for read permission.	<b>W_OK</b>	Test for write permission.	<b>X_OK</b>	Test for execute (search) permission.	<b>F_OK</b>	Test for existence of file.	<b>NULL</b>	Null pointer	<b>F_ULOCK</b>	Unlock a previously locked region.	<b>F_LOCK</b>	Lock a region for exclusive use.	<b>F_TLOCK</b>	Test and lock a region for exclusive use.	<b>F_TEST</b>	Test a region for other processes locks.	<b>SEEK_SET</b>	Set file offset to <i>offset</i> .	<b>SEEK_CUR</b>	Set file offset to current plus <i>offset</i> .	<b>SEEK_END</b>	Set file offset to EOF plus <i>offset</i> .	<b>_POSIX_VERSION</b>	Integer value indicating version of the POSIX standard (see <b>standards(5)</b> ).	<b>_XOPEN_VERSION</b>	integer value indicating version of the XPG to which system conforms.	<b>_POSIX_JOB_CONTROL</b>	Implementation supports job control.	<b>_POSIX_SAVED_IDS</b>	The exec functions (see <b>exec(2)</b> ) save the effective user and group.
<b>R_OK</b>	Test for read permission.																																
<b>W_OK</b>	Test for write permission.																																
<b>X_OK</b>	Test for execute (search) permission.																																
<b>F_OK</b>	Test for existence of file.																																
<b>NULL</b>	Null pointer																																
<b>F_ULOCK</b>	Unlock a previously locked region.																																
<b>F_LOCK</b>	Lock a region for exclusive use.																																
<b>F_TLOCK</b>	Test and lock a region for exclusive use.																																
<b>F_TEST</b>	Test a region for other processes locks.																																
<b>SEEK_SET</b>	Set file offset to <i>offset</i> .																																
<b>SEEK_CUR</b>	Set file offset to current plus <i>offset</i> .																																
<b>SEEK_END</b>	Set file offset to EOF plus <i>offset</i> .																																
<b>_POSIX_VERSION</b>	Integer value indicating version of the POSIX standard (see <b>standards(5)</b> ).																																
<b>_XOPEN_VERSION</b>	integer value indicating version of the XPG to which system conforms.																																
<b>_POSIX_JOB_CONTROL</b>	Implementation supports job control.																																
<b>_POSIX_SAVED_IDS</b>	The exec functions (see <b>exec(2)</b> ) save the effective user and group.																																

**\_POSIX\_VDISABLE** Terminal special characters defined in `<termios.h>` (see `termio(7I)`) can be disabled using this character.

The following symbolic constants are defined for `sysconf(3C)`:

**\_SC\_ARG\_MAX**  
**\_SC\_CLK\_TCK**  
**\_SC\_JOB\_CONTROL**  
**\_SC\_NGROUPS\_MAX**  
**\_SC\_OPEN\_MAX**  
**\_SC\_PAGESIZE**  
**\_SC\_PASS\_MAX**  
**\_SC\_SAVED\_IDS**  
**\_SC\_VERSION**  
**\_SC\_XOPEN\_VERSION**

The following symbolic constants are defined for `fpathconf(2)`:

**\_PC\_CHOWN\_RESTRICTED**  
**\_PC\_LINK\_MAX**  
**\_PC\_MAX\_CANON**  
**\_PC\_MAX\_INPUT**  
**\_PC\_NAME\_MAX**  
**\_PC\_NO\_TRUNC**  
**\_PC\_PATH\_MAX**  
**\_PC\_PIPE\_BUF**  
**\_PC\_VDISABLE**

The following symbolic constants are defined for file streams:

**STDIN\_FILENO** File number (0) of `stdin`.  
**STDOUT\_FILENO** File number (1) of `stdout`.  
**STDERR\_FILENO** File number (2) of `stderr`.

The following pathnames are defined:

**GF\_PATH** Pathname of the group file.  
**PF\_PATH** Pathname of the passwd file.

The following values for constants are defined:

**\_POSIX\_VERSION** 199009L  
**\_XOPEN\_VERSION** 3

**SEE ALSO** `access(2)`, `exec(2)`, `fcntl(2)`, `fpathconf(2)`, `lseek(2)`, `lockf(3C)`, `sysconf(3C)`, `termios(3)`, `group(4)`, `passwd(4)`, `standards(5)`, `termio(7I)`

<b>NAME</b>	updaters – configuration file for NIS updating
<b>SYNOPSIS</b>	<code>/var/yp/updaters</code>
<b>DESCRIPTION</b>	<p>The file <code>/var/yp/updaters</code> is a makefile (see <b>make(1S)</b>) which is used for updating the Network Information Service (NIS) databases. Databases can only be updated in a secure network, that is, one that has a <b>publickey(4)</b> database. Each entry in the file is a make target for a particular NIS database. For example, if there is an NIS database named <b>passwd.byname</b> that can be updated, there should be a <b>make</b> target named <b>passwd.byname</b> in the <b>updaters</b> file with the command to update the file.</p> <p>The information necessary to make the update is passed to the update command through standard input. The information passed is described below (all items are followed by a NEWLINE except for 4 and 6):</p> <ol style="list-style-type: none"> <li>1. Network name of client wishing to make the update (a string).</li> <li>2. Kind of update (an integer).</li> <li>3. Number of bytes in key (an integer).</li> <li>4. Actual bytes of key.</li> <li>5. Number of bytes in data (an integer).</li> <li>6. Actual bytes of data.</li> </ol> <p>After receiving this information through standard input, the command to update the particular database determines whether the user is allowed to make the change. If not, it exits with the status <b>YPERR_ACCESS</b>. If the user is allowed to make the change, the command makes the change and exits with a status of zero. If there are any errors that may prevent the <b>updaters</b> from making the change, it should exit with the status that matches a valid NIS error code described in <code>&lt;rpcsvc/ypclnt.h&gt;</code>.</p>
<b>FILES</b>	<code>/var/yp/updaters</code> The makefile used for updating the NIS databases.
<b>SEE ALSO</b>	<b>make(1S)</b> , <b>rpc.yupdated(1M)</b> , <b>publickey(4)</b>
<b>NOTES</b>	The Network Information Service (NIS) was formerly known as Sun Yellow Pages (YP). The functionality of the two remains the same; only the name has changed. The name Yellow Pages is a registered trademark in the United Kingdom of British Telecommunications plc, and may not be used without permission.

**NAME** utmp, wtmp – utmp and wtmp entry formats

**SYNOPSIS** `#include <utmp.h>`

**DESCRIPTION** `utmp` and `wtmp` hold user and accounting information for commands such as `who`, `write`, and `login`. These files have the following structure, defined in `<utmp.h>`:

```

#define UTMP_FILE          "/var/adm/utmp"
#define WTMP_FILE          "/var/adm/wtmp"
#define ut_name            ut_user

struct utmp {
    char ut_user[8];        /* user login name */
    char ut_id[4];         /* /sbin/inittab id (created by */
                           /* process that puts entry in utmp) */
    char ut_line[12];      /* device name (console, lxxx) */
    short ut_pid;          /* process id */
    short ut_type;         /* type of entry */
    struct exit_status {
        short e_termination; /* process termination status */
        short e_exit;        /* process exit status */
    } ut_exit;             /* exit status of a process
                           /* marked as DEAD_PROCESS */
    time_t ut_time;        /* time entry was made */
};
/* Definitions for ut_type */
#define EMPTY            0
#define RUN_LVL          1
#define BOOT_TIME        2
#define OLD_TIME         3
#define NEW_TIME         4
#define INIT_PROCESS     5/* process spawned by "init" */
#define LOGIN_PROCESS    6/* a "getty" process waiting for login */
#define USER_PROCESS     7/* a user process */
#define DEAD_PROCESS     8
#define ACCOUNTING      9
#define UTMAXTYPE        ACCOUNTING/* max legal value of ut_type */
/* Below are special strings or formats used in the "ut_line" */
/* field when accounting for something other than a process. */
/* No string for the ut_line field can be more than 11 chars + */
/* a null character in length. */
#define RUNLVL_MSG       "run-level %c"
#define BOOT_MSG         "system boot"
#define OTIME_MSG        "old time"
#define NTIME_MSG        "new time"

```

**FILES** /var/adm/utmp  
/var/adm/wtmp

**SEE ALSO** login(1), who(1), write(1)

**NAME** utmpx, wtmpx – utmpx and wtmpx entry formats

**SYNOPSIS** `#include <utmpx.h>`

**DESCRIPTION** `utmpx` is an extended version of `utmp(4)`.  
`utmpx` and `wtmpx` hold user and accounting information for commands such as `who`, `write`, and `login`. These files have the following structure as defined by `<utmpx.h>`:

```

#define UTMPX_FILE          "/var/adm/utmpx"
#define WTMPX_FILE          "/var/adm/wtmpx"
#define ut_name             ut_user
#define ut_xtime            ut_tv.tv_sec
struct utmpx {
    char    ut_user[32];      /* user login name */
    char    ut_id[4];        /* inittab id */
    char    ut_line[32];     /* device name */
                                /* (console, lnx) */
    pid_t   ut_pid;          /* process id */
    short   ut_type;        /* type of entry */
    struct  exit_status ut_exit; /* process termination/exit */
                                /* status */
    struct  timeval ut_tv;   /* time entry was made */
    long    ut_session;     /* session ID, used for */
                                /* windowing */
    long    pad[5];         /* reserved for future use */
    short   ut_syslen;      /* significant length of */
                                /* ut_host */
                                /* including terminating null */
    char    ut_host[257];    /* remote host name */
};
/* Definitions for ut_type */
#define EMPTY             0
#define RUN_LVL           1
#define BOOT_TIME        2
#define OLD_TIME         3
#define NEW_TIME         4
#define INIT_PROCESS     5      /* Process spawned by "init" */
#define LOGIN_PROCESS    6      /* A "getty" process waiting */
                                /* for login */
#define USER_PROCESS     7      /* A user process */
#define DEAD_PROCESS     8
#define ACCOUNTING       9

#define UTMXTYPE ACCOUNTING /* Largest legal value */
                                /* of ut_type */

```

```
/* Below are special strings or formats used in the "ut_line" */  
/* field when accounting for something other than a process. */  
/* No string for the ut_line field can be more than 11 chars + */  
/* a null character in length. */
```

```
#define RUNLVL_MSG          "run-level %c"  
#define BOOT_MSG           "system boot"  
#define OTIME_MSG         "old time"  
#define NTIME_MSG         "new time"  
#define MOD_WIN            10
```

**FILES** /var/adm/utmpx  
/var/adm/wtmpx

**SEE ALSO** login(1), who(1), write(1), utmp(4)



<b>NAME</b>	vfstab – table of file system defaults							
<b>DESCRIPTION</b>	<p>The file <b>/etc/vfstab</b> describes defaults for each file system. The information is stored in a table with the following column headings:</p> <table border="0"> <tr> <td><i>device to mount</i></td> <td><i>device to fsck</i></td> <td><i>mount point</i></td> <td><i>FS type</i></td> <td><i>fsck pass</i></td> <td><i>mount at boot</i></td> <td><i>mount options</i></td> </tr> </table> <p>The fields in the table are space-separated and show the resource name (<i>device to mount</i>), the raw device to <b>fsck</b> (<i>device to fsck</i>), the default mount directory (<i>mount point</i>), the name of the file system type (<i>FS type</i>), the number used by <b>fsck</b> to decide whether to check the file system automatically (<i>fsck pass</i>), whether the file system should be mounted automatically by <b>mountall</b> (<i>mount at boot</i>), and the file system mount options (<i>mount options</i>). (See respective mount file system man page below in <b>SEE ALSO</b> for <i>mount options</i>.) A '-' is used to indicate no entry in a field. This may be used when a field does not apply to the resource being mounted.</p> <p>The <b>getvfsent(3C)</b> family of routines is used to read and write to <b>/etc/vfstab</b>.</p> <p><b>/etc/vfstab</b> may be used to specify swap areas. An entry so specified, (which can be a file or a device), will automatically be added as a swap area by the <b>/sbin/swapadd</b> script when the system boots. To specify a swap area, the <i>device-to-mount</i> field contains the name of the swap file or device, the <i>FS-type</i> is "swap", <i>mount-at-boot</i> is "no" and all other fields have no entry.</p>	<i>device to mount</i>	<i>device to fsck</i>	<i>mount point</i>	<i>FS type</i>	<i>fsck pass</i>	<i>mount at boot</i>	<i>mount options</i>
<i>device to mount</i>	<i>device to fsck</i>	<i>mount point</i>	<i>FS type</i>	<i>fsck pass</i>	<i>mount at boot</i>	<i>mount options</i>		
<b>SEE ALSO</b>	<p><b>fsck(1M)</b>, <b>mount(1M)</b>, <b>mount_cachefs(1M)</b>, <b>mount_hfs(1M)</b>, <b>mount_nfs(1M)</b>, <b>mount_tmpfs(1M)</b>, <b>mount_ufs(1M)</b>, <b>setmnt(1M)</b>, <b>swap(1M)</b>, <b>getvfsent(3C)</b></p> <p><i>System Administration Guide</i></p>							

<b>NAME</b>	vme – configuration files for VMEbus device drivers														
<b>DESCRIPTION</b>	<p>Some Solaris platforms support the VMEbus as a peripheral expansion bus to allow VME devices to be connected to the system. Drivers for these devices need to use driver configuration files to inform the system that the device hardware may be present. The configuration file also must specify the device addresses on the VMEbus and any interrupt capabilities that the device may have.</p> <p>Configuration files for VMEbus device drivers should identify the parent bus driver implicitly using the <i>class</i> keyword. This removes the dependency on the name of the particular bus driver involved since this may be named differently on different platforms. See <b>driver.conf(4)</b> for further details of configuration file syntax.</p> <p>All bus drivers of class <b>vme</b> recognise the following properties:</p> <p><b>reg</b>           An arbitrary length array where each element of the array consists of a 3-tuple of integers. Each array element describes a logically contiguous mappable resource on the VMEbus.</p> <p>The first integer of the tuple specifies the type of access. The value is derived from the size of transfer and the address modifier bits used to access the locations. The table below shows the values used for common VME devices accessed in supervisor mode:</p> <table border="1" data-bbox="896 816 1208 1060"> <thead> <tr> <th>Address space</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td><b>A16D16</b></td> <td><b>0x2d</b></td> </tr> <tr> <td><b>A24D16</b></td> <td><b>0x3d</b></td> </tr> <tr> <td><b>A32D16</b></td> <td><b>0xd</b></td> </tr> <tr> <td><b>A16D32</b></td> <td><b>0x6d</b></td> </tr> <tr> <td><b>A24D32</b></td> <td><b>0x7d</b></td> </tr> <tr> <td><b>A32D32</b></td> <td><b>0x4d</b></td> </tr> </tbody> </table> <p>The second integer of each 3-tuple specifies the offset in the address space identified by the first element. The third integer of each 3-tuple specifies the size, in bytes, of the mappable region.</p> <p>The driver can refer to the elements of this array by index, and construct kernel mappings to these addresses using <b>ddi_map_regs(9F)</b>. The index into the array is passed as the <i>rnumber</i> argument of <b>ddi_map_regs()</b>.</p> <p><b>interrupts</b>   An arbitrary length array where each element of the array consists of a pair of integers. Each array element describes a possible interrupt that the device might generate.</p> <p>The first integer of each pair specifies the VMEbus interrupt level. The second integer of each pair specifies the VMEbus vector number. The driver can refer to the elements of this array by index, and register interrupt handlers with the system using <b>ddi_add_intr(9F)</b>. The index into the array is passed as the <i>inumber</i> argument of <b>ddi_add_intr()</b>.</p>	Address space	Value	<b>A16D16</b>	<b>0x2d</b>	<b>A24D16</b>	<b>0x3d</b>	<b>A32D16</b>	<b>0xd</b>	<b>A16D32</b>	<b>0x6d</b>	<b>A24D32</b>	<b>0x7d</b>	<b>A32D32</b>	<b>0x4d</b>
Address space	Value														
<b>A16D16</b>	<b>0x2d</b>														
<b>A24D16</b>	<b>0x3d</b>														
<b>A32D16</b>	<b>0xd</b>														
<b>A16D32</b>	<b>0x6d</b>														
<b>A24D32</b>	<b>0x7d</b>														
<b>A32D32</b>	<b>0x4d</b>														

All VMEbus device drivers must provide **reg** properties. The first two integer elements of this property are used to construct the address part of the device name under **/devices**.

Only devices that generate interrupts need to provide **interrupts** properties.

**EXAMPLES**

Here is a configuration file called **SUNW,diskctrl.conf** for a VMEbus disk controller card called **SUNW,diskctrl**.

The device provides two sets of registers, both should be accessed with supervisor accesses and the A16D32 address modifier bits (16 bits of address, 32 bit data transfers). Both registers occupy 32 bytes; one register set starts at address 0xee80, the other is at 0xef00. The device can generate interrupts at VME level 2 with a VME vector number of 0x92.

```
#
# Copyright (c) 1992, by Sun Microsystems, Inc.
#ident "@(#)SUNW,diskctrl.conf 1.4 92/05/11 SMI"
#

name="SUNW,diskctrl" class="vme"
      reg=0x6d,0xee80,32,0x6d,0xef00,32
      interrupts=2,0x92;
```

**ATTRIBUTES**

See **attributes(5)** for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Architecture	SPARC

**SEE ALSO**

**driver.conf(4)**, **attributes(5)**, **ddi\_add\_intr(9F)**, **ddi\_map\_regs(9F)**, **ddi\_prop\_op(9F)**

*Writing Device Drivers*

*ANSI/IEEE Std 1014-1987: IEEE Standard for a Versatile Backplane Bus: VMEbus*

<b>NAME</b>	vold.conf – Volume Management configuration file
<b>SYNOPSIS</b>	<i>/etc/vold.conf</i>
<b>DESCRIPTION</b>	<p>The <b>vold.conf</b> file contains the Volume Management configuration information used by <b>vold(1M)</b>. This information includes the database to use, labels that are supported, devices to use, actions to take when certain media events occur, and the list of file systems that are unsafe to eject without unmounting.</p> <p>Modify <b>vold.conf</b> to specify which program should be called when media events happen (actions) or when you need to add another device to your system. See the example section for more information on adding devices.</p> <p>If you modify <b>vold.conf</b>, you must tell <b>vold</b> to reread <b>vold.conf</b> by sending a HUP signal. Use</p> <pre style="margin-left: 2em;"># ps -ef   grep vold # kill -HUP vold_pid</pre>
<b>File Format</b>	<p>The syntax for the <b>vold.conf</b> file is shown here.</p> <pre style="margin-left: 2em;"># Database to use db database  # Labels supported label label_type shared_object device  # Devices to use use device type special shared_object symname [ options ]  # Actions insert regex [ options ] program program args eject regex [ options ] program program args notify regex [ options ] program program args  # List of file system types unsafe to eject unsafe fs_type fs_type</pre> <p>Of these syntax fields, you can safely modify <b>Devices to use</b> and <b>Actions</b>.</p>
<b>Devices to Use Field</b>	<p>All <b>use device</b> statements must be grouped together by device type. (For example, all <b>use cdrom</b> statements must be grouped together; and all <b>use floppy</b> statements must be grouped together.) Here are the explanations of the syntax for the <b>Devices to use</b> field.</p> <p><i>device</i>                   The type of removable media device to be used. Legal values are <b>cdrom</b> and <b>floppy</b>.</p> <p><i>type</i>                     The specific capabilities of the device. Legal value is <b>drive</b>.</p> <p><i>special</i>                 This <b>sh(1)</b> expression specifies the device or devices to be used. Path usually begins with <b>/dev</b>.</p>

*shared\_object* The name of the program that manages this device. **vold(1M)** expects to find this program in **/usr/lib/vold**.

*symname* The symbolic name that refers to this device. The *symname* is placed in the device directory.

*options* The user, group, and mode permissions for the media inserted (optional).

The *special* and *symname* parameters are related. If *special* contains any shell wildcard characters (i.e., has one or more asterisks or question marks in it), then the *symname* must have a "%d" at its end. In this case, the devices that are found to match the regular expression are sorted, then numbered. The first device will have a zero filled in for the "%d", the second device found will have a one, and so on.

If the *special* specification does not have any shell wildcard characters then the *symname* parameter must explicitly specify a number at its end (see **EXAMPLES** below).

**Actions Field**

Here are the explanations of the syntax for the **Actions** field.

**insert | eject | notify** The media event prompting the event

*regex* This **sh(1)** regular expression is matched against each entry in the **/vol** file system that is being affected by this event.

*options* You can specify what user or group name that this event is to run as (optional).

*program* The full path name of an executable program to be run when *regex* is matched.

*program args* Arguments to the program.

**Default Values**

The default **vold.conf** file is shown here.

```
#
# Volume Daemon Configuration file
#

# Database to use (must be first)
db db_mem.so

# Labels supported
label dos label_dos.so floppy
label cdrom label_cdrom.so cdrom
label sun label_sun.so floppy

# Devices to use
use cdrom drive /dev/dsk/c*s2 dev_cdrom.so cdrom%d
use floppy drive /dev/diskette[0-9] dev_floppy.so floppy%d
```

```
# Actions
insert /vol*/dev/fd[0-9]/* user=root /usr/sbin/rmmount
insert /vol*/dev/dsk/* user=root /usr/sbin/rmmount
eject /vol*/dev/fd[0-9]/* user=root /usr/sbin/rmmount
eject /vol*/dev/dsk/* user=root /usr/sbin/rmmount
notify /vol*/rdsk/* group=tty user=root /usr/lib/vold/volmissing -p
```

```
# List of file system types unsafe to eject
unsafe ufs hsfs pcfs
```

**EXAMPLES**

To add a CD-ROM drive to the **vold.conf** file that does not match the default regular expression (**/dev/rdsk/c\*s2**), you must explicitly list its device path and what symbolic name (with **%d**) you want the device path to have. For example, to add a CD-ROM drive that has the path **/dev/rdsk/my/cdroms?** (where **s?** are the different slices), add the following line to **vold.conf** (all on one line):

```
use cdrom drive /dev/rdsk/my/cdroms2 dev_cdrom.so cdrom%d
```

Then, when a volume is inserted in this CD-ROM drive, volume management will assign it the next symbolic name. For example, if two CD-ROMs match the default regular expression, they would be named **cdrom0** and **cdrom1**; and any that match the added regular expression would be named starting with **cdrom2**.

For a diskette that does not match the **vold.conf** default regular expression (**/dev/floppy[0-9]**), a similar line would have to be added for the diskette. For example, to add a diskette whose path was **/dev/my/fd0**, you would add the following to **vold.conf**:

```
use floppy drive /dev/my/fd0 dev_floppy.so floppy%d
```

**SEE ALSO**

**sh(1)**, **volcancel(1)**, **volcheck(1)**, **volmissing(1)**, **rmmount(1M)**, **vold(1M)**, **rmmount.conf(4)**, **volfs(7FS)**

**NOTES**

Volume Management manages both the block and character device for CD-ROMs and floppy disks; but, to make the configuration file easier to set up and scan, only one of these devices needs to be specified. If you follow the conventions specified below, Volume Management figures out both device names if only one of them is specified. For example, if you specify the block device, it figures out the pathname to the character device; if you specify the pathname to the character device, it figures out the block device.

**CD-ROM Naming Conventions**

The CD-ROM pathname must have a directory component of **rdsk** (for the character device) and **dsk** for the block device. For example, if you specify the character device using the line:

```
use cdrom drive /dev/rdsk/my/cdroms2 dev_cdrom.so cdrom%d
```

then it is assumed that the block device is at

```
/dev/dsk/my/cdroms2
```

**Floppy Disk Naming  
Conventions**

For floppy disks, Volume Management requires that the device pathnames end in either **rfd**[0-9] or **rdiskette**[0-9] for the character device, and **fd**[0-9] or **diskette**[0-9] for the block device. As with the CD-ROM, it generates either the block name given the character name, or the character name given the block name.

<b>NAME</b>	ypfiles – Network Information Service Version 2, formerly known as YP																												
<b>DESCRIPTION</b>	<p>The NIS network information service uses a distributed, replicated database of <b>dbm</b> files (in ASCII form) contained in the <b>/var/yp</b> directory hierarchy on each NIS server. NIS has been replaced by NIS+, the new version of the Network Information Service. See <b>nis+(1)</b>. This release only supports the client functionality of NIS, (see <b>ypclnt(3N)</b>). The client functions are either supported by the <b>ypserv</b> process running on a machine with an earlier version of SunOS or by the NIS+ server in "YP-compatibility" mode, (see <b>rpc.nisd(1M)</b>).</p> <p>A <b>dbm</b> database served by the NIS server is called an NIS <i>map</i>. An NIS <i>domain</i> is a sub-directory of <b>/var/yp</b> containing a set of NIS maps on each NIS server.</p> <p>Standard nicknames are defined in the file <b>/var/yp/nicknames</b>. These names can be used in place of the full map name in the <b>ypmatch</b> and <b>ypcat</b> commands. The command <b>ypwhich -m</b> can be used to display the full set of nicknames. Each line of the nickname file contains two fields separated by white space. The first field is the nickname and the second field is the name of the map that it expands to. The nickname cannot contain a ".".</p>																												
<b>FILES</b>	<b>/var/yp/nicknames</b> nicknames file																												
<b>SEE ALSO</b>	<b>nis+(1)</b> , <b>nisaddent(1M)</b> , <b>nissetup(1M)</b> , <b>rpc.nisd(1M)</b> , <b>ypbind(1M)</b> , <b>ypinit(1M)</b> , <b>dbm(3B)</b> , <b>secure_rpc(3N)</b> , <b>ypclnt(3N)</b>																												
<b>NOTES</b>	<p>The NIS+ server, <b>rpc.nisd</b>, when run in "YP-compatibility mode", can support NIS clients only for the standard NIS maps listed below, provided that it has been set up to serve the corresponding NIS+ tables using <b>nissetup(1M)</b> and <b>nisaddent(1M)</b>. The NIS+ server should serve the directory with the same name (case sensitive) as the domainname of the NIS client. NIS+ servers use secure RPC to verify client credentials but the NIS clients do not authenticate their requests using secure RPC. Therefore, NIS clients can look up the information stored by the NIS+ server only if the information has "read" access for an unauthenticated client (i.e. one with "nobody" NIS+ credentials).</p> <table border="0" style="margin-left: 40px;"> <thead> <tr> <th style="text-align: left;"><i>NIS maps</i></th> <th style="text-align: left;"><i>NIS+ tables</i></th> </tr> </thead> <tbody> <tr> <td><b>passwd.byname</b></td> <td><b>passwd.org_dir</b></td> </tr> <tr> <td><b>passwd.byuid</b></td> <td><b>passwd.org_dir</b></td> </tr> <tr> <td><b>group.byname</b></td> <td><b>group.org_dir</b></td> </tr> <tr> <td><b>group.bygid</b></td> <td><b>group.org_dir</b></td> </tr> <tr> <td><b>publickey.byname</b></td> <td><b>cred.org_dir</b></td> </tr> <tr> <td><b>hosts.byaddr</b></td> <td><b>hosts.org_dir</b></td> </tr> <tr> <td><b>hosts.byname</b></td> <td><b>hosts.org_dir</b></td> </tr> <tr> <td><b>mail.byaddr</b></td> <td><b>mail_aliases.org_dir</b></td> </tr> <tr> <td><b>mail.aliases</b></td> <td><b>mail_aliases.org_dir</b></td> </tr> <tr> <td><b>services.byname</b></td> <td><b>services.org_dir</b></td> </tr> <tr> <td><b>services.byservicename</b></td> <td><b>services.org_dir</b></td> </tr> <tr> <td><b>rpc.bynumber</b></td> <td><b>rpc.org_dir</b></td> </tr> <tr> <td><b>rpc.byname</b></td> <td><b>rpc.org_dir</b></td> </tr> </tbody> </table>	<i>NIS maps</i>	<i>NIS+ tables</i>	<b>passwd.byname</b>	<b>passwd.org_dir</b>	<b>passwd.byuid</b>	<b>passwd.org_dir</b>	<b>group.byname</b>	<b>group.org_dir</b>	<b>group.bygid</b>	<b>group.org_dir</b>	<b>publickey.byname</b>	<b>cred.org_dir</b>	<b>hosts.byaddr</b>	<b>hosts.org_dir</b>	<b>hosts.byname</b>	<b>hosts.org_dir</b>	<b>mail.byaddr</b>	<b>mail_aliases.org_dir</b>	<b>mail.aliases</b>	<b>mail_aliases.org_dir</b>	<b>services.byname</b>	<b>services.org_dir</b>	<b>services.byservicename</b>	<b>services.org_dir</b>	<b>rpc.bynumber</b>	<b>rpc.org_dir</b>	<b>rpc.byname</b>	<b>rpc.org_dir</b>
<i>NIS maps</i>	<i>NIS+ tables</i>																												
<b>passwd.byname</b>	<b>passwd.org_dir</b>																												
<b>passwd.byuid</b>	<b>passwd.org_dir</b>																												
<b>group.byname</b>	<b>group.org_dir</b>																												
<b>group.bygid</b>	<b>group.org_dir</b>																												
<b>publickey.byname</b>	<b>cred.org_dir</b>																												
<b>hosts.byaddr</b>	<b>hosts.org_dir</b>																												
<b>hosts.byname</b>	<b>hosts.org_dir</b>																												
<b>mail.byaddr</b>	<b>mail_aliases.org_dir</b>																												
<b>mail.aliases</b>	<b>mail_aliases.org_dir</b>																												
<b>services.byname</b>	<b>services.org_dir</b>																												
<b>services.byservicename</b>	<b>services.org_dir</b>																												
<b>rpc.bynumber</b>	<b>rpc.org_dir</b>																												
<b>rpc.byname</b>	<b>rpc.org_dir</b>																												



<b>protocols.bynumber</b>	<b>protocols.org_dir</b>
<b>protocols.byname</b>	<b>protocols.org_dir</b>
<b>networks.byaddr</b>	<b>networks.org_dir</b>
<b>networks.byname</b>	<b>networks.org_dir</b>
<b>netmasks.bymask</b>	<b>netmasks.org_dir</b>
<b>netmasks.byaddr</b>	<b>netmasks.org_dir</b>
<b>ethers.byname</b>	<b>ethers.org_dir</b>
<b>ethers.byaddr</b>	<b>ethers.byname</b>
<b>bootparams</b>	<b>bootparams</b>
<b>auto.master</b>	<b>auto_master.org_dir</b>
<b>auto.home</b>	<b>auto_home.org_dir</b>
<b>auto.direct</b>	<b>auto_direct.org_dir</b>
<b>auto.src</b>	<b>auto_src.org_dir</b>

# Index

---

## *Special Characters*

- .clustertoc — listing of software packages on product distribution media, 4-49
- .environ — user-preference variables files for AT&T FACE, 4-93
- .order — installation order of software packages on product distribution media, 4-236
- .packagetoc — listing of software packages on product distribution media, 4-238
- .pref — user-preference variables files for AT&T FACE, 4-93
- .variables — user-preference variables files for AT&T FACE, 4-93
- /proc, the process file system — `proc`, 4-286

## **A**

- a.out — Executable and Linking (ELF) files, 4-20
- accounting files
  - `acct`, 4-12
  - `utmp`, 4-449
  - `utmpx`, 4-451
  - `wtmp`, 4-449
  - `wtmpx`, 4-451
- accounting system
  - prime/nonprime hours — `holidays`, 4-109
- `acct` — process accounting file format, 4-12
- addresses — addresses for sendmail, 4-17

- `admin` — installation defaults file, 4-14
- aliases — sendmail aliases file, 4-17
- an alternative memory allocator library — `libmap-  
malloc`, 4-159
- `ar` — archive file format, 4-22
- archive file format — `ar`, 4-22
- archives — device header, 4-25
- ASET environment file — `asetenv`, 4-28
- ASET master files
  - `asetmasters`, 4-30
  - `cklist.high`, 4-30
  - `cklist.low`, 4-30
  - `cklist.med`, 4-30
  - `tune.high`, 4-30
  - `tune.low`, 4-30
  - `tune.med`, 4-30
  - `uid_aliases`, 4-30
- `asetenv` — ASET environment file, 4-28
- `audit` — audit control file, 4-34, 4-37
- audit trail file
  - `audit.log`, 4-39
- `audit.log` — audit trail file, 4-39
- `audit_class` password file, 4-32
- `audit_event` password file, 4-38
- `audit_user` password file, 4-43

---

## B

basic security library — `libbssm`, 4-134  
boot parameter database — `bootparams`, 4-44  
BOOTP  
    network database — `dhcp_network`, 4-70  
`bootparams` — boot parameter database, 4-44

## C

C library — `libc`, 4-135  
CD-ROM table of contents file — `cdtoc`, 4-46  
`cdtoc` — CD-ROM table of contents file, 4-46  
compatible versions file — `compver`, 4-53  
`compver` — compatible versions file, 4-53  
configuration file for default router(s) — `default-trouter`, 4-59  
configuration file, system log daemon — `syslogd`, 4-368  
connect accounting  
    — `wtmp`, 4-449  
    — `wtmpx`, 4-451  
copyright — copyright information file, 4-54  
core — core image of a terminated process file, 4-55

## D

`d_passwd` — dial-up password file, 4-88  
    Generating An Encrypted Password, 4-89  
database parameters for DHCP — `dhcp`, 4-69  
`default_fs` — specify the default file system type for local or remote file systems, 4-58  
`defaultrouter` — configuration file for default router(s), 4-59  
`depend` — software dependencies file, 4-60  
devconfig configuration files — `device.cfinfo`, 4-63  
device id library — `libdevvid`, 4-150  
device instance number file — `path_to_inst`, 4-252  
`device.cfinfo` — devconfig configuration files, 4-63  
`device_allocate`  
    device access control file, 4-61

`device_maps`  
    device access control file, 4-67  
devices  
    access control file — `device_allocate`, 4-61, 4-67  
devices, capabilities  
    terminal and printers — `terminfo`, 4-379  
dfs utilities packages  
    list — `fstypes`, 4-103  
`dfstab` — file containing commands for sharing resources, 4-68  
parameters for DHCP databases — `dhcp`, 4-69  
    client identifier to IP address mappings — `dhcp_network`, 4-70  
    configuration parameter table — `dhcptab`, 4-73  
DHCP option mnemonic mapping table — `dhcptags`, 4-81  
`dhcp_network`, also see `pntadm`, 4-70  
`dhcptab` — DHCP configuration parameter table, 4-73  
`dhcptags` — DHCP option mnemonic mapping table, 4-81  
dial-up password file — `d_passwd`, 4-88  
`dialups` — list of terminal devices requiring a dial-up password, 4-85  
`dir_ufs` — format of ufs directories, 4-87  
directory of files specifying supported platforms — `platform`, 4-265  
`dirent` — file system independent directory entry, 4-86  
disk drive configuration for the format command — `format.dat`, 4-99  
disk space requirement file — `space`, 4-361  
dispatcher, real-time process  
    parameters — `rt_dptbl`, 4-340  
dispatcher, time-sharing process  
    parameters — `ts_dptbl`, 4-432  
`driver.conf` — driver configuration file, 4-90  
drivers  
    driver for EISA devices — `eisa`, 4-363  
    driver for ISA devices — `isa`, 4-363  
    driver for MCA devices — `mca`, 4-363  
    driver for PCI devices — `pci`, 4-254

---

drivers, *continued*

- driver for pseudo devices — `pseudo`, 4-317
- driver for SBus devices — `vme`, 4-346
- driver for SCSI devices — `scsi`, 4-352
- driver for VME devices — `vme`, 4-454

dynamic linking interface library — `libdl`, 4-151

## E

- `eisa` — configuration file for EISA bus device drivers, 4-363
- ELF access library — `libelf`, 4-154
- ELF files — `a.out`, 4-20
- `environ` — user-preference variables files for AT&T FACE, 4-93
- environment
  - setting up an environment for user at login time — `profile`, 4-310
- `ethers` — Ethernet addresses of hosts on Internet, 4-95
- Executable and Linking Format (ELF) files — `a.out`, 4-20

## F

- FACE
  - alias file — `pathalias`, 4-251
  - object architecture information — `ott`, 4-237
- FACE object architecture information — `ott`, 4-237
- `fd` — file descriptor files, 4-96
- File Access Control List library — `libsec`, 4-176
- file descriptor files — `fd`, 4-96
- file formats
  - `intro`, 4-5
- file system
  - defaults — `vfstab`, 4-453
  - mounted — `mtab`, 4-210
- file that maps sockets to transport providers — `sock2path`, 4-360
- `filehdr` — file header for common object files, 4-97
- files used by programs
  - `/etc/security/device_maps` — `device_maps` file, 4-67

files used by programs, *continued*

- `/etc/security/device_allocate` — `device_allocate` file, 4-62
- format of a font file used as input to the `loadfont` utility — `loadfont`, 4-202
- format of a ufs file system volume — `fs_ufs`, 4-104
  - `inode`, 4-104
  - `inode_ufs`, 4-104
- `format.dat` — disk drive configuration for the `format` command, 4-99
  - Keywords, 4-99
  - Syntax, 4-99
- `forward` — mail forwarding file, 4-17
- `fs_ufs` — format of a ufs file system volume, 4-104
- `fspec` — format specification in text files, 4-102
- `fstypes` — file that lists utilities packages for distributed file system, 4-103

## G

- general administrative library — `libadm`, 4-132
- graphics interface files — `plot`, 4B-269
- `group` — local source of group information, 4-107

## H

- `holidays` — prime/nonprime hours for accounting system, 4-109
- host name database — `hosts`, 4-110
- `hosts` — host name data base, 4-110
- `hosts.equiv` — trusted hosts list, 4-112

## I

- `inetd.conf` — Internet server database, 4-115
- `init.d` — initialization and termination scripts for changing init states, 4-117
- initialization and termination scripts for changing init states — `init.d`, 4-117
- `inittab` — script for `init`, 4-118
- `inode` — format of a ufs file system volume, 4-104
- `inode_ufs` — format of a ufs file system volume, 4-104
- installation

---

installation, *continued*

defaults file — admin, 4-14

internationalization library — libintl, 4-155

Internet

DHCP database — dhcp\_network, 4-70

Ethernet addresses of hosts — ethers, 4-95

network name database — networks, 4-223

protocol name database — protocols, 4-312

services and aliases — services, 4-355

Internet servers database — servers, 4-115

ioctl for sockets

SIOCADDRT — add route, 4-335

SIOCDELRT — delete route, 4-335

isa — configuration file for ISA bus device drivers, 4-363

issue — issue identification file, 4-121

## K

Kerberos configuration file

— krb.conf, 4-130

Kerberos library — libkrb, 4-156

Kerberos realm translation file

— krb.realms, 4-131

kernel packet forwarding database — route, 4-331

kernel statistics library — libkstat, 4-157

Kernel Virtual Memory access library — libkvm, 4-158

keyboard table descriptions for loadkeys and dumpkeys — keytables, 4-122

keytables — keyboard table descriptions for loadkeys and dumpkeys, 4-122

## L

legal annotations

specify — note, 4-227

libadm — general administrative library, 4-132

libaio — the asynchronous I/O library, 4-133

libbsm — basic security library, 4-134

libc — the C library, 4-135

libci — Sun Solstice Enterprise Agent Component Interface Library, 4-149

libdevd — device id library, 4-150

libdl — the dynamic linking interface library,

4-151

libdmi — Sun Solstice Enterprise Agent DMI Library, 4-152

libdmimi — Sun Solstice Enterprise Agent Management Interface Library, 4-153

libelf — ELF access library, 4-154

libintl — internationalization library, 4-155

libkrb — Kerberos library, 4-156

libkstat — kernel statistics library, 4-157

libkvm — Kernel Virtual Memory access library, 4-158

/usr/lib/libkvm.so.1, 4-158

libmapmalloc — an alternative memory allocator library, 4-159

libmp — multiple precision library, 4-160

libnisdb — NIS+ Database access library, 4-161

libnsl — the network services library, 4-162

/usr/lib/libnsl.so.1, 4-162

libposix4 — POSIX.1b Realtime Extensions library, 4-168

libpthread — POSIX threads library, 4-169

/usr/lib/libpthread.so.1, 4-169

librac — remote asynchronous calls library, 4-171

library  
C library — libc, 4-135

dynamic linking interface library — libdl, 4-151

library file format — ar, 4-22

libresolv — resolver library, 4-172

librpcsoc — obsolete RPC library, 4-174

librpcsvc — Miscellaneous RPC services library, 4-175

libsec — File Access Control List library, 4-176

/usr/lib/libsec.so.1, 4-176

libsocket — the sockets library, 4-177

/usr/lib/libsocket.so.1, 4-177

libssagent — Sun Solstice Enterprise Agent Library, 4-179

libssasmp — Sun Solstice Enterprise SNMP Library, 4-180

libsys — the system library, 4-181

/usr/lib/libc.so.1, 4-181

Index-4

libthread — the threads library, 4-185  
     /usr/lib/libthread.so.1, 4-185  
 libthread\_db — threads debugging library, 4-188  
     /usr/lib/libthread\_db.so.1, 4-188  
 libucb — the UCB compatibility library, 4-190  
 libvolmgt — volume management library, 4-192  
 libw — the wide character library, 4-193  
 libxfn — the XFN interface library, 4-195  
     /usr/lib/libxfn.so.1, 4-195  
 libxnet — X/Open Networking Interfaces library,  
     4-198  
 limits — header for implementation-specific con-  
     stants, 4-200  
 link editor output — a.out, 4-20  
 list of network groups — netgroup, 4-215  
 list of terminal devices requiring a dial-up pass-  
     word — dialups, 4-85  
 loadfont — format of a font file used as input to  
     the loadfont utility, 4-202  
 login-based device permissions — logindevperm,  
     4-206  
 logindevperm — login-based device permissions,  
     4-206  
 loginlog — log of failed login attempts, 4-207

## M

magic — file command's magic numbers table,  
     4-208  
 mca — configuration file for MCA bus device  
     drivers, 4-363  
 message displayed to users attempting to log on in  
     the process of a system shutdown — nolo-  
     gin, 4-226  
 Miscellaneous RPC services library — librpcsvc,  
     4-175  
 mounted file system table — mtab, 4-210  
 mtab — mounted file system table, 4-210  
 multiple precision library — libmp, 4-160

## N

name servers  
     configuration file — resolv.conf, 4-324  
 name service cache daemon configuration —  
     nscd.conf  
     nscd.conf, 4-228  
 name service switch  
     configuration file — nsswitch.conf, 4-230  
 netconfig — network configuration database,  
     4-211  
 netgroup — list of network groups, 4-215  
 netgroup — list of network groups, 4-215  
 netid — netname database, 4-217  
 netmasks — network masks for subnetting, 4-219  
 netname database — netid, 4-217  
     .netrc — ftp remote login data file, 4-221  
 Network Information Service Version 2, formerly  
     known as YP — ypfiles, 4-460  
 network packet routing device — routing, 4-335  
 networks connected to the system — netconfig,  
     4-211  
 networks — network name database, 4-223  
 NFS  
     remote monted file systems — rmtab, 4-330  
 NIS databases  
     updating — updaters, 4-448  
 NIS+ Database access library — libnisdb, 4-161  
 nisfiles — NIS+ database files and directory  
     structure, 4-224  
 nologin — message displayed to users attempting  
     to log on in the process of a system shutdown ,  
     4-226  
 nonprime hours  
     accounting system — holidays, 4-109  
 note — specify legal annotations, 4-227  
 nscd.conf — name service cache daemon  
     configuration, 4-228  
 nsswitch.conf — configuration file for the name  
     service switch, 4-230

---

## O

### object files

file header — `filehdr`, 4-97

obsolete RPC library — `librpcsoc`, 4-174

## P

### package characteristics file

— `pkginfo`, 4-259

### package contents description file

— `pkgmap`, 4-262

### package information file — prototype, 4-313

### package installation order file

— `order`, 4-236

### package table of contents description file

`.clustertoc` — `clustertoc`, 4-49

— `packagetoc`, 4-238

### packet routing device — routing, 4-335

### packet routing ioctls

`SIOCADDRT` — add route, 4-335

`SIOCDELRT` — delete route, 4-335

### packing rules file for cachefs and filesync — pack-

`ingrules`, 4-242

### packingrules — packing rules file for cachefs

and filesync, 4-242

### `pam.conf` — configuration file for pluggable

authentication modules, 4-245

### `passwd` — password file, 4-249

### passwords

access-restricted shadow system file — `shadow`, 4-356

### `path_to_inst` — device instance number file, 4-252

### `pathalias` — alias file for FACE, 4-251

### PCI devices

driver class — `pci`, 4-254

### `pci` — drivers for PCI devices, 4-254

### `pcmcia` — PCMCIA nexus driver, 4-257

### PCMCIA nexus driver — `pcmcia`, 4-257

### phones — remote host phone numbers, 4-258

### `pkginfo` — software package characteristics file, 4-259

### `pkgmap` — listing of software package contents, 4-262

### `platform` — directory of files specifying supported platforms, 4-265

### `plot` — graphics interface files, 4B-269

### POSIX threads library — `libpthread`, 4-169

### POSIX.1b Realtime Extensions library — `libposix4`, 4-168

### power management configuration file —

`power.conf`, 4-271

### `power.conf` — power management configuration file, 4-271

### prime hours

accounting system — `holidays`, 4-109

### printers — printer alias database, 4-277

### `printers.conf` — printing configuration database, 4-280

### `proc` — `/proc`, the process file system, 4-286

`PCAGENT`, 4-305

`PCCFAULT`, 4-302

`PCCSIG`, 4-301

`PCKILL`, 4-301

`PCNICE`, 4-306

`PCREAD PCWRITE`, 4-306

`PCRUN`, 4-300

`PCSCRED`, 4-306

`PCSENTRY PCSEXIT`, 4-302

`PCSET PCUNSET`, 4-304

`PCSFAULT`, 4-301

`PCSFREG`, 4-305

`PCSHOLD`, 4-301

`PCSREG`, 4-305

`PCSSIG`, 4-301

`PCSTOP PCDSTOP PCWSTOP PCTWSTOP`, 4-299

`PCSTRACE`, 4-300

`PCSVADDR`, 4-305

`PCSXREG`, 4-305

`PCUNKILL`, 4-301

`PCWATCH`, 4-302

### process accounting

— `acct`, 4-12

### process file system — `proc`, 4-286

### process scheduler (or dispatcher), real-time

parameters — `rt_dptbl`, 4-340

### process scheduler (or dispatcher), time-sharing

---

process scheduler (or dispatcher), time-sharing, *continued*  
parameters — `ts_dptbl`, 4-432

processes  
core image of a terminated process file —  
core, 4-55

profile — setting up an environment for user at  
login time, 4-310

project identification file — `issue`, 4-121

protocols — names of known protocols in Inter-  
net, 4-312

prototype — package information file, 4-313

pseudo devices, 4-317

pseudo — drivers for pseudo devices, 4-317

publickey — publickey database for secure RPC,  
4-318

## Q

queuedefs — queue description file for at, batch,  
and cron spooled by at or batch or atrm, 4-319

## R

real-time process dispatcher  
parameters — `rt_dptbl`, 4-340

real-time process scheduler  
parameters — `rt_dptbl`, 4-340

remote asynchronous calls library — `librac`, 4-171

remote authentication for hosts and users —  
`hosts.equiv`, `.rhosts`, 4-112

remote — remote host descriptions, 4-321

remote host  
phone numbers — `phones`, 4-258

remote login data for ftp — `netrc`, 4-221

remote mounted file systems  
— `rmtab`, 4-330

Remote Program Load (RPL) server configuration  
file — `rpld.conf`, 4-338

`resolv.conf` — configuration file for name server  
routines, 4-324

resolver library — `libresolv`, 4-172

`rmmount.conf` — removable media moun-  
ter configuration file

`rmmount.conf` — removable media moun-  
ter configuration file, *continued*  
Default Values, 4-327  
Examples, 4-327

`route` — kernel packet forwarding database, 4-331  
Messages, 4-332

`routing` — local network packet routing, 4-335

`routing ioctls`  
`SIOCADDRT` — add route, 4-335  
`SIOCDELRT` — delete route, 4-335

`rpc` — rpc program number database, 4-337

RPC program names  
for program numbers — `rpc`, 4-337

RPC security  
public key database — `publickey`, 4-318

`rpld.conf` — Remote Program Load (RPL) server  
configuration file, 4-338

## S

SBus devices  
driver class — `sbus`, 4-346

`sbus` — drivers for SBus devices, 4-346

`sccsfile` — format of SCCS history file, 4-349

scheduler, real-time process  
parameters — `rt_dptbl`, 4-340

scheduler, time-sharing process  
parameters — `ts_dptbl`, 4-432

SCSI devices  
driver class — `scsi`, 4-352

`scsi` — drivers for SCSI devices, 4-352

`securenets` file, 4-354

`sendmail` addresses file — `addresses`, 4-17

`sendmail` aliases file — `aliases`, 4-17

`sendmail` aliases file — `forward`, 4-17

`services` — Internet services and aliases, 4-355

`shadow` password file, 4-356

share resources across network, commands —  
`dfstab`, 4-68

shared resources, local  
— `sharetab`, 4-358

`sharetab` — shared file system table, 4-358

shell database — `shells`, 4-359

`shells` — shell database, 4-359



---

sock2path — file that maps sockets to transport providers, 4-360

software dependencies — depend, 4-60

space — disk space requirement file, 4-361

specify the default file system type for local or remote file systems — default\_fs, 4-58

su command log file — sulog, 4-362

sulog — su command log file, 4-362

Sun Solstice Enterprise Agent Component Interface Library — libci, 4-149

Sun Solstice Enterprise Agent DMI Library — libdmi, 4-152

Sun Solstice Enterprise Agent Library — libssagent, 4-179

Sun Solstice Enterprise Agent Management Interface Library — libdmimi, 4-153

Sun Solstice Enterprise SNMP Library — libssasmp, 4-180

symbolic constants

- header — unistd, 4-446

sysbus — configuration files for ISA, EISA, and MCA bus device drivers, 4-363

- eisa, 4-363
- isa, 4-363
- mca, 4-363

sysidcfg — system identification configuration file, 4-365

- Keyword Syntax Rules, 4-365
- Keywords, 4-366
- Where To Put the sysidcfg File, 4-365

syslogd.conf — system log daemon configuration file, 4-368

system — system configuration information, 4-371

system identification configuration file — sysidcfg, 4-365

system log configuration file — syslogd.conf, 4-368

## T

telnet default options file — telnetrc, 4-375

telnetrc — file for telnet default options, 4-375

term — format of compiled term file, 4-376

terminals

- line setting information — ttydefs, 4-439

termination and initialization scripts for changing init states — init.d, 4-117

terminfo — System V terminal capability data base, 4-379

test files

- format specification — fspec, 4-102

the asynchronous I/O library — libaio, 4-133

the network services library — libnsl, 4-162

the sockets library — libsocket, 4-177

the system library — libsys, 4-181

the threads library — libthread, 4-185

the UCB compatibility library — libucb, 4-190

the wide character library — libw, 4-193

the XFN interface library — libxfn, 4-195

threads debugging library — libthread\_db, 4-188

timezone — set default time zone, 4-424

time-sharing process dispatcher

- parameters — ts\_dptbl, 4-432

time-sharing process scheduler

- parameters — ts\_dptbl, 4-432

timed event services

- queue description file for at, batch and cron — queuedefs, 4-319

timezone — default timezone data base, 4-425

TNF kernel probes — tnf\_kernel\_probes, 4-426

tnf\_kernel\_probes — TNF kernel probes, 4-426

ttydefs — terminal line settings information, 4-439

ttyname

- list of directories with terminal-related device files — ttysrch, 4-440

## U

ufs

- format — dir\_ufs, 4-87

ufsdump — incremental dump format, 4-442

unistd — header for symbolic constants, 4-446

updaters — configuration file for NIS updating, 4-448

user-preference variables files for AT&T FACE —

---

environ, 4-93  
utmp — format for utmp file, 4-449  
utmp — format for utmpx file, 4-451

## V

vfstab — defaults for each file system, 4-453  
VME devices  
  driver class — vme, 4-454  
vme — drivers for VME devices, 4-454  
vold.conf — Volume Management configuration  
  file, 4-456  
  Actions Field, 4-457  
  CD-ROM Naming Conventions, 4-458  
  Default Values, 4-457  
  Devices to Use Field, 4-456  
  File Format, 4-456  
  Floppy Disk Naming Conventions, 4-459  
Volume Management  
  configuration file — vold.conf, 4-456  
volume management library — libvolmgt, 4-192

## W

wtmp — format for wtmp file, 4-449  
wtmp — format for wtmpx file, 4-451

## X

X/Open Networking Interfaces library — libx-  
net, 4-198

## Y

ypfiles — Network Information Service Version  
2, formerly known as YP, 4-460