



man pages section 3: Library Interfaces and Headers

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 817-0713-10
April 2004

Copyright 2004 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 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, docs.sun.com, AnswerBook, AnswerBook2, and Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

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

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

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 2004 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

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

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

Sun, Sun Microsystems, le logo Sun, docs.sun.com, AnswerBook, AnswerBook2, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

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

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A 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.



040116@7518



Contents

Preface 7

Introduction 13

Intro(3) 14

Library Interfaces and Headers 27

acct(3HEAD) 28

aio(3HEAD) 30

ar(3HEAD) 31

dirent(3HEAD) 34

fcntl(3HEAD) 35

floatingpoint(3HEAD) 39

in(3HEAD) 41

inet(3HEAD) 43

langinfo(3HEAD) 44

libadm(3LIB) 47

libaio(3LIB) 48

libbsdmalloc(3LIB) 49

libbsm(3LIB) 50

libc(3LIB) 52

libcfgadm(3LIB) 79

libcpc(3LIB) 80

libcrypt(3LIB) 81

libcurses(3LIB) 82

libcurses(3LIBUCB) 89

libdbm(3LIBUCB) 91
libdevid(3LIB) 92
libdevinfo(3LIB) 93
libdl(3LIB) 95
libdmi(3LIB) 96
libdmici(3LIB) 97
libdmimi(3LIB) 98
libefi(3LIB) 99
libelf(3LIB) 100
libexacct(3LIB) 102
libform(3LIB) 104
libgen(3LIB) 106
libgss(3LIB) 108
libidnkit(3LIB) 110
libintl(3LIB) 113
libkstat(3LIB) 114
libkvm(3LIB) 115
libl(3LIB) 116
liblayout(3LIB) 117
liblgrp(3LIB) 118
libmail(3LIB) 119
libmalloc(3LIB) 120
libmapmalloc(3LIB) 121
libmd5(3LIB) 122
libmenu(3LIB) 123
libmp(3LIB) 125
libmtmalloc(3LIB) 126
libnsl(3LIB) 127
libnvpair(3LIB) 135
libpam(3LIB) 137
libpanel(3LIB) 139
libpctx(3LIB) 140
libpicl(3LIB) 141
libpicltree(3LIB) 142
libplot(3LIB) 144
libpool(3LIB) 146
libproject(3LIB) 150
libpthread(3LIB) 151

librac(3LIB)	154
libresolv(3LIB)	155
librpcsoc(3LIBUCB)	157
librpcsvc(3LIB)	158
librsm(3LIB)	159
librt(3LIB)	161
librtld_db(3LIB)	163
libsec(3LIB)	164
libsecdb(3LIB)	165
libsndfile(3LIB)	167
libslp(3LIB)	168
libsmartcard(3LIB)	169
libsocket(3LIB)	171
libssagent(3LIB)	173
libssasmp(3LIB)	174
libsys(3LIB)	175
libsysevent(3LIB)	180
libtermcap(3LIBUCB)	181
libthread(3LIB)	182
libthread_db(3LIB)	186
libtnfctl(3LIB)	188
libucb(3LIBUCB)	190
libumem(3LIB)	192
libvolmgt(3LIB)	193
libw(3LIB)	194
libwsreg(3LIB)	196
libxfn(3LIB)	198
libxnet(3LIB)	203
liby(3LIB)	206
math(3HEAD)	207
mqueue(3HEAD)	208
ndbm(3HEAD)	209
netdb(3HEAD)	210
nl_types(3HEAD)	214
sched(3HEAD)	215
siginfo(3HEAD)	216
signal(3HEAD)	220
socket(3HEAD)	227

stat(3HEAD) 232
stdarg(3HEAD) 234
time(3HEAD) 236
types32(3HEAD) 238
types(3HEAD) 239
ucontext(3HEAD) 241
un(3HEAD) 242
unistd(3HEAD) 243
values(3HEAD) 250
varargs(3HEAD) 252
wstat(3HEAD) 254

Index 255

Preface

Both novice users and those familiar with the SunOS operating system can use online man pages to obtain information about the system and its features. 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.

Overview

The following contains a brief description of each man page section 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.
- 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.
- 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 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 can 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 path name is shown. 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:
[]	Brackets. The option or argument enclosed in these brackets is optional. If the brackets are omitted, the argument must be specified.
. . .	Ellipses. Several values can 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 a 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.
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, and functions are described under USAGE.
IOCTL	This section appears on pages in Section 7 only. Only the device class that supplies appropriate parameters to the <code>ioctl(2)</code> system call is called <code>ioctl</code> and generates its own heading. <code>ioctl</code> calls for a specific device are listed alphabetically (on the man page for that specific device). <code>ioctl</code> calls are used for a particular class of devices all of which have an <code>io</code> ending, such as <code>mtio(7I)</code> .
OPTIONS	This section 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 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 <code>errno</code> 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 lists special rules, features, and commands that require in-depth explanations. The subsections listed here 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 <code>example%</code> , or if the user must be superuser, <code>example#</code> . Examples are followed by explanations, variable substitution rules, or returned values. Most examples illustrate concepts from the SYNOPSIS, DESCRIPTION, OPTIONS, and USAGE sections.
ENVIRONMENT VARIABLES	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 file names 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 <code>attributes(5)</code> 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.
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.

Introduction

Intro(3)

NAME	Intro – introduction to functions and libraries
DESCRIPTION	<p>This section describes functions found in various Solaris libraries, other than those functions described in Section 2 of this manual that directly invoke UNIX system primitives. Function declarations can be obtained from the <code>#include</code> files indicated on each page. Pages are grouped by library and are identified by the library name (or an abbreviation of the library name) after the section number. Collections of related libraries are grouped into five volumes as described below. A sixth volume (listed first) contains pages describing the contents of each shared library and each header used by the functions, macros, and external variables described in the remaining five volumes.</p>
Library Interfaces and Headers	<p>This volume describes the contents of each shared library and each header used by functions, macros, and external variables described in the remaining five volumes.</p> <p>(3LIB) The libraries described in this section are implemented as shared objects.</p> <p> Descriptions of shared objects may include a definition of the global symbols that define the shared objects' public interface, for example <code>SUNW_1.1</code>. Other interfaces may exist within the shared object, for example <code>SUNW_private.1.1</code>. 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 on 32-bit systems only. 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> <p>(3LIBUCB) The SunOS/BSD Compatibility libraries described in this section are implemented as a shared object. See (3LIB) above.</p> <p>(3HEAD) The headers described in this section are used by functions, macros, and external variables. Headers contain function prototypes, definitions of symbolic constants, common structures, preprocessor macros, and defined types. Each function described in the remaining five volumes specifies the headers that an application must include in order to use that function. In most cases only one header is required. These headers are present on an application development system; they do have to be present on the target execution system.</p>
Basic Library Functions	<p>The functions described in this volume are the core C library functions that are basic to application development.</p> <p>(3C) These functions, together with those of Section 2, constitute the standard C library, <code>libc</code>, which is automatically linked by the C</p>

compilation system. The standard C library is implemented as a shared object, `libc.so`, and as an archive, `libc.a`. C programs are linked with the shared object version of the standard C library by default. Specify `-Bstatic` or `-dn` on the `cc` command line to link with the archive version. See `libc(3LIB)`, `cc(1B)` for other overrides, and the “C Compilation System” chapter of the *ANSI C Programmer’s Guide* for a discussion. Some functions behave differently in standard-conforming environments. This behavior is noted on the individual manual pages. See `standards(5)`.

- (3DL) These functions constitute the dynamic linking library, `libdl`. This library is implemented as a shared object, `libdl.so`, but is not automatically linked by the C compilation system. Specify `-ldl` on the `cc` command line to link with this library. See `libdl(3LIB)`.
- (3MALLOC) These functions constitute the various memory allocation libraries: `libmalloc`, `libbsdmalloc`, `libmapmalloc`, `libmtmalloc`, and `libumem`. Each of these libraries is implemented as a shared object (`libmalloc.so`, `libbsdmalloc.so`, `libmapmalloc.so`, `libmtmalloc.so`, and `libumem.so`) and `libmalloc`, `libbsdmalloc`, and `libmapmalloc` are implemented as archives (`libmalloc.a`, `libbsdmalloc.a`, and `libmapmalloc.a`). These libraries are not automatically linked by the C compilation system. Specify `-lmalloc`, `-lbsdmalloc`, `-lmapmalloc`, `-lmtmalloc`, and `-lumem` to link with, respectively, `libmalloc`, `libbsdmalloc`, `libmapmalloc`, `libmtmalloc`, and `libumem`. See `libmalloc(3LIB)`, `libbsdmalloc(3LIB)`, `libmapmalloc(3LIB)`, `libmtmalloc(3LIB)`, and `libumem(3LIB)`.
- (3UCB) These functions constitute the Source Compatibility (with BSD functions) library. It is implemented as a shared object, `libucb.so`, and as an archive, `libucb.a`, but is not automatically linked by the C compilation system. Specify `-lucb` on the `cc` command line to link with this library, which is located in the `/usr/ucb` subdirectory. Headers for this library are located within `/usr/ucbinclude`. See `libucb(3LIBUCB)`.

Networking Library Functions

The functions described in this volume comprise the various networking libraries.

- (3GSS) The functions in this library are the routines that comprise the Generic Security Services API library. This library is implemented as a shared object, `libgss.so.1`, but it is not automatically linked by the C compilation system. Specify `-lgss` on the `cc` command line to link with this library. See `libgss(3LIB)`.
- (3LDAP) These functions constitute the Lightweight Directory Access Protocol library, `libldap`. This library is implemented as a shared object, `libldap.so`, but is not automatically linked by the C

Intro(3)

	<p>compilation system. Specify <code>-lldap</code> on the <code>cc</code> command line to link with this library. See <code>ldap(3LDAP)</code>.</p>
(3NSL)	<p>These functions constitute the Network Service Library, <code>libnsl</code>. This library is implemented as a shared object, <code>libnsl.so</code>, and as an archive, <code>libnsl.a</code>, but is not automatically linked by the C compilation system. Specify <code>-lnsl</code> on the <code>cc</code> command line to link with this library. See <code>libnsl(3LIB)</code>.</p>
	<p>Many base networking functions are also available in the X/Open Networking Interfaces library, <code>libxnet</code>. See section (3XNET) below for more information on the <code>libxnet</code> interfaces.</p>
(3RAC)	<p>These functions constitute the remote asynchronous calls library, <code>librac</code>. This library is implemented as a shared object, <code>librac.so</code>, and as an archive, <code>librac.a</code>, but is not automatically linked by the C compilation system. Specify <code>-lrac</code> on the <code>cc</code> command line to link with this library. See <code>librac(3LIB)</code>.</p>
(3RESOLV)	<p>These functions constitute the resolver library, <code>libresolv</code>. This library is implemented as a shared object, <code>libresolv.so</code>, and as an archive, <code>libresolv.a</code>, but is not automatically linked by the C compilation system. Specify <code>-lresolv</code> on the <code>cc</code> command line to link with this library. See <code>libresolv(3LIB)</code>.</p>
(3RPC)	<p>These functions constitute the remote procedure call libraries, <code>librpcsvc</code> and <code>librpcsoc</code>. The latter is provided for compatibility only; new applications should not link to it. Both libraries are implemented as shared objects, <code>librpcsvc.so</code> and <code>librpcsoc.so</code>, respectively, and <code>librpcsvc</code> is implemented as an archive, <code>librpcsvc.a</code>. Neither library is automatically linked by the C compilation system. Specify <code>-lrpcsvc</code> or <code>-lrpcsoc</code> on the <code>cc</code> command line to link with these libraries. See <code>librpcsvc(3LIB)</code> and <code>librpcsoc(3LIBUCB)</code>.</p>
(3SLP)	<p>These functions constitute the Service Location Protocol library, <code>libslp</code>. This library is implemented as a shared object, <code>libslp.so.1</code>, but it is not automatically linked by the C compilation system. See <code>libslp(3LIB)</code></p>
(3SOCKET)	<p>These functions constitute the sockets library, <code>libsocket</code>. This library is implemented as a shared object, <code>libsocket.so</code>, and as an archive, <code>libsocket.a</code>, but is not automatically linked by the C compilation system. Specify <code>-lsocket</code> on the <code>cc</code> command line to link with this library. See <code>libsocket(3LIB)</code>.</p>
(3XFN)	<p>These functions constitute the X/Open Federated Naming library, <code>libxfn</code>. This library is implemented as a shared object, <code>libxfn.so</code>, but is not automatically linked by the C compilation</p>

system. Specify `-lxfn` on the `cc` command line to link with this library. See `libxfn(3LIB)`, `xfn(3XFN)`, `fns(5)`, and `standards(5)`.

(3XNET)

These functions constitute X/Open networking interfaces which comply with the X/Open CAE Specification, Networking Services, Issue 4 (September, 1994). This library is implemented as a shared object, `libxnet.so`, but is not automatically linked by the C compilation system. Specify `-lxnet` on the `cc` command line to link with this library. See `libxnet(3LIB)` and `standards(5)` for compilation information.

Under all circumstances, the use of the Sockets API is recommended over the XTI and TLI APIs. If portability to other XPGV4v2 (see `standards(5)`) systems is a requirement, the application must use the `libxnet` interfaces. If portability is not required, the sockets interfaces in `libsocket` and `libnsl` are recommended over those in `libxnet`. Between the XTI and TLI APIs, the XTI interfaces (available with `libxnet`) are recommended over the TLI interfaces (available with `libnsl`).

Curses Library Functions

The functions described in this volume comprise the libraries that provide graphics and character screen updating capabilities.

(3CURSES)

The functions constitute the following libraries:

`libcurses` These functions constitute the curses library, `libcurses`. This library is implemented as a shared object, `libcurses.so`, and as an archive, `libcurses.a`, but is not automatically linked by the C compilation system. Specify `-lcurses` on the `cc` command line to link with this library. See `libcurses(3LIB)`.

`libform` These functions constitute the forms library, `libform`. This library is implemented as a shared object, `libform.so`, and as an archive, `libforms.a`, but is not automatically linked by the C compilation system. Specify `-lform` on the `cc` command line to link with this library. See `libform(3LIB)`.

`libmenu` These functions constitute the menus library, `libmenu`. This library is implemented as a shared object, `libmenu.so`, and as an archive, `libmenu.a`, but is not automatically linked by the C compilation system. Specify `-lmenu` on the `cc` command line to link with this library. See `libmenu(3LIB)`.

`libpanel` These functions constitute the panels library, `libpanel`. This library is implemented as a

Intro(3)

		shared object, <code>libpanel.so</code> , and as an archive, <code>libpanel.a</code> , but is not automatically linked by the C compilation system. Specify <code>-lpanel</code> on the <code>cc</code> command line to link with this library. See <code>libpanel(3LIB)</code> .
(3PLOT)		These functions constitute the graphics library, <code>libplot</code> . This library is implemented as a shared object, <code>libplot.so</code> , and as an archive, <code>libplot.a</code> , but is not automatically linked by the C compilation system. Specify <code>-lplot</code> on the <code>cc</code> command line to link with this library. See <code>libplot(3LIB)</code> .
(3XCURSES)		These functions constitute the X/Open Curses library, located in <code>/usr/xpg4/lib/libcurses.so.1</code> . This library provides a set of internationalized functions and macros for creating and modifying input and output to a terminal screen. Included in this library are functions for creating windows, highlighting text, writing to the screen, reading from user input, and moving the cursor. X/Open Curses is designed to optimize screen update activities. The X/Open Curses library conforms fully with Issue 4 of the X/Open Extended Curses specification.
Threads and Realtime Library Functions		The functions described in this volume constitute the threads and realtime libraries.
(3AIO)		These functions constitute the asynchronous I/O library, <code>liaio</code> . This library is implemented as a shared object, <code>libaio.so</code> , but is not automatically linked by the C compilation system. Specify <code>-laio</code> on the <code>cc</code> command line to link with this library. See <code>libaio(3LIB)</code> .
(3DOOR)		These functions constitute the doors library, <code>libdoor</code> . This library is implemented as a shared object, <code>libdoor.so</code> , but is not automatically linked by the C compilation system. Specify <code>-ldoor</code> on the <code>cc</code> command line to link with this library.
(3RT)		These functions constitute the POSIX.4 Realtime library, <code>librt</code> . It is implemented as a shared object, <code>librt.so</code> , but is not automatically linked by the C compilation system. Specify <code>-lrt</code> on the <code>cc</code> command line to link with this library. Note that the former name for this library, <code>libposix4</code> , is maintained for backward compatibility but should be avoided. See <code>librt(3LIB)</code> .
(3SCHED)		These functions constitute the LWP scheduling library, <code>libsched</code> . This library is implemented as a shared object, <code>libsched.so</code> , but is not automatically linked by the C compilation system. Specify <code>-lsched</code> on the <code>cc</code> command line to link with this library. .
(3THR)		These functions constitute the threads libraries, <code>libpthread</code> , <code>libthread</code> , and <code>libthread_db</code> . The <code>libpthread</code> and <code>libthread</code> libraries are used for building multithreaded

applications: `libpthread` implements the POSIX (see `standards(5)`) threads interface, whereas `libthread` implements the Solaris threads interface. The `libthread_db` library is useful for building debuggers for multithreaded applications.

Both POSIX threads and Solaris threads can be used within the same application. Their implementations are completely compatible with each other; however, only POSIX threads guarantee portability to other POSIX-conforming environments.

When POSIX and Solaris threads are used in the same application, if there are calls with the same name but different semantics, the POSIX semantic supersedes the Solaris threads semantic. For example, the call to `fork()` will imply the `fork1()` semantic in a program linked with the POSIX threads library, whether or not it is also linked with `-lthread` (Solaris threads).

The `libpthread`, `libthread`, and `libthread_db` libraries are implemented as shared objects, `libpthread.so`, `libthread_db.so`, and `libthread.so`, respectively. These libraries are not automatically linked by the C compilation system. Specify `-lpthread`, `-lthread`, or `-lthread_db` on the `cc` command line to link with these libraries. See `libpthread(3LIB)`, `libthread(3LIB)`, and `libthread_db(3LIB)`.

Extended Library Functions

The functions described in this volume comprise various specialized libraries that are not limited to the following:

- (3BSM) These functions constitute the basic security library, `libbsm`. This library is implemented as a shared object, `libbsm.so`, and as an archive, `libbsm.a`, but is not automatically linked by the C compilation system. Specify `-lbsm` on the `cc` command line to link with this library. See `libbsm(3LIB)`.
- (3CFGADM) These functions constitute the configuration administration library, `libcfgadm`. This library is implemented as a shared object, `libcfgadm.so`, but is not automatically linked by the C compilation system. Specify `-lcfgadm` on the `cc` command line to link with this library. See `libcfgadm(3LIB)`.
- (3CPC) These functions constitute the CPU performance counter library, `libcpc`, and the process context library, `libpctx`. These libraries are implemented as shared objects, `libcpc.so` and `libpctx.so`, respectively, but are not automatically linked by the C compilation system. Specify `-lcpc` or `-lpctx` on the `cc` command line to link with these libraries. See `libcpc(3LIB)` and `libpctx(3LIB)`.
- (3DEVID) These functions constitute the device ID library, `libdevid`. This library is implemented as a shared object, `libdevid.so`, but is

Intro(3)

	<p>not automatically linked by the C compilation system. Specify <code>-ldevld</code> on the <code>cc</code> command line to link with this library. See <code>libdevld(3LIB)</code>.</p>
(3DEVINFO)	<p>These functions constitute the device information library, <code>libdevinfo</code>. This library is implemented as a shared object, <code>libdevinfo.so</code>, but is not automatically linked by the C compilation system. Specify <code>-ldevinfo</code> on the <code>cc</code> command line to link with this library. See <code>libdevinfo(3LIB)</code>.</p>
(3DMI)	<p>These functions constitute the DMI libraries, <code>libdmi</code>, <code>libdmici</code>, and <code>libdmimi</code>. These libraries are implemented as shared objects, <code>libdmi.so</code>, <code>libdmici.so</code>, and <code>libdmimi.so</code>, respectively, but are not automatically linked by the C compilation system. Specify <code>-ldmi</code>, <code>-ldmici</code>, or <code>-ldmimi</code> on the <code>cc</code> command line to link with these libraries. See <code>libdmi(3LIB)</code>, <code>libdmici(3LIB)</code>, and <code>libdmimi(3LIB)</code>.</p>
(3ELF)	<p>These functions constitute the ELF access library, <code>libelf</code>, (Extensible Linking Format). This library provides the interface for the creation and analyses of “elf” files; executables, objects, and shared objects. <code>libelf</code> is implemented as a shared object, <code>libelf.so</code>, and as an archive, <code>libelf.a</code>, but is not automatically linked by the C compilation system. Specify <code>-lelf</code> on the <code>cc</code> command line to link with this library. See <code>libelf(3LIB)</code>.</p>
(3EXACCT)	<p>These functions constitute the extended accounting access library, <code>libexacct</code>, and the project database access library, <code>libproject</code>. These libraries are implemented as shared objects, <code>libexacct.so</code> and <code>libproject.so</code>, respectively, but are not automatically linked by the C compilation system. Specify <code>-lexacct</code> or <code>-lproject</code> on the <code>cc</code> command line to link with these libraries. See <code>libexacct(3LIB)</code> and <code>libproject(3LIB)</code>.</p>
(3GEN)	<p>These functions constitute the string pattern-matching and pathname manipulation library, <code>libgen</code>. This library is implemented as a shared object, <code>libgen.so</code>, and as an archive, <code>libgen.a</code>, but is not automatically linked by the C compilation system. Specify <code>-lgen</code> on the <code>cc</code> command line to link with this library. See <code>libgen(3LIB)</code>.</p>
(3KSTAT)	<p>These functions constitute the kernel statistics library, which is implemented as a shared object, <code>libkstat.so</code>, and as an archive, <code>libkstat.a</code>, but is not automatically linked by the C compilation system. Specify <code>-lkstat</code> on the <code>cc</code> command line to link with this library. See <code>libkstat(3LIB)</code>.</p>
(3KVM)	<p>These functions allow access to the kernel’s virtual memory library, which is implemented as a shared object, <code>libkvm.so</code>, and</p>

	as an archive, <code>libkvm.a</code> , but is not automatically linked by the C compilation system. Specify <code>-lkvm</code> on the <code>cc</code> command line to link with this library. See <code>libkvm(3LIB)</code> .
(3LAYOUT)	These functions constitute the layout service library, which is implemented as a shared object, <code>liblayout.so</code> , but is not automatically linked by the C compilation system. Specify <code>-llayout</code> on the <code>cc</code> command line to link with this library. See <code>liblayout(3LIB)</code> .
(3LGRP)	These functions constitute the locality group library, which is implemented as a shared object, <code>liblgrp.so</code> , but is not automatically linked by the C compilation system. Specify <code>-llgrp</code> on the <code>cc</code> command line to link with this library. See <code>liblgrp(3LIB)</code> .
(3M)	These functions constitute the mathematical library, <code>libm</code> . This library is implemented as a shared object, <code>libm.so</code> , and as an archive, <code>libm.a</code> , but is not automatically linked by the C compilation system. Specify <code>-lm</code> on the <code>cc</code> command line to link with this library.
(3MAIL)	These functions constitute the user mailbox management library, <code>libmail</code> . This library is implemented as a shared object, <code>libmail.so</code> , and as an archive, <code>libmail.a</code> , but is not automatically linked by the C compilation system. Specify <code>-lmail</code> on the <code>cc</code> command line to link with this library.
(3MP)	These functions constitute the integer mathematical library, <code>libmp</code> . This library is implemented as a shared object, <code>libmp.so</code> , and as an archive, <code>libmp.a</code> , but is not automatically linked by the C compilation system. Specify <code>-lmp</code> on the <code>cc</code> command line to link with this library. See <code>libmp(3LIB)</code> .
(3NVPAIR)	These functions constitute the name–value pair library, <code>libnvpair</code> . This library is implemented as a shared object, <code>libnvpair.so</code> , but is not automatically linked by the C compilation system. Specify <code>-lnvpair</code> on the <code>cc</code> command line to link with this library. See <code>libnvpair(3LIB)</code> .
(3PAM)	These functions constitute the Pluggable Authentication Module (PAM) library, <code>libpam</code> . This library is implemented as a shared object, <code>libpam.so</code> , and as an archive, <code>libpam.a</code> , but is not automatically linked by the C compilation system. Specify <code>-lpam</code> on the <code>cc</code> command line to link with this library. See <code>libpam(3LIB)</code> .
(3PICL)	These functions constitute the PICL library, <code>libpicl</code> . This library is implemented as a shared object, <code>libpicl.so</code> , but is not automatically linked by the C compilation system. Specify <code>-lpicl</code>

Intro(3)

	on the <code>cc</code> command line to link with this library. See <code>libpicl(3LIB)</code> and <code>libpicl(3PICL)</code> .
(3PICLTREE)	These functions constitute the PICL plug-in library, <code>libpicltree</code> . This library is implemented as a shared object, <code>libpicltree.so</code> , but is not automatically linked by the C compilation system. Specify <code>-lpicltree</code> on the <code>cc</code> command line to link with this library. See <code>libpicltree(3LIB)</code> and <code>libpicltree(3PICLTREE)</code> .
(3POOL)	These functions constitute the pool configuration manipulation library, <code>libpool</code> . This library is implemented as a shared object, <code>libpool.so</code> , but is not automatically linked by the C compilation system. Specify <code>-lpool</code> on the <code>cc</code> command line to link with this library. See <code>libpool(3LIB)</code> .
(3PROJECT)	These functions constitute the project database access library, <code>libproject</code> . This library is implemented as a shared object, <code>libproject.so</code> , but is not automatically linked by the C compilation system. Specify <code>-lproject</code> on the <code>cc</code> command line to link with this library. See <code>libproject(3LIB)</code> .
(3RSM)	These functions constitute the remote shared memory library, <code>librsm</code> . This library is implemented as a shared object, <code>librsm.so</code> , but is not automatically linked by the C compilation system. Specify <code>-lrsm</code> on the <code>cc</code> command line to link with this library. See <code>librsm(3LIB)</code> .
(3SEC)	These functions constitute the file access control library, <code>libsec</code> . This library is implemented as a shared object, <code>libsec.so</code> , and as an archive, <code>libsec.a</code> , but is not automatically linked by the C compilation system. Specify <code>-lsec</code> on the <code>cc</code> command line to link with this library. See <code>libsec(3LIB)</code> .
(3SECDB)	These functions constitute the security attributes database library, <code>libsecdb</code> . This library is implemented as a shared object, <code>libsecdb.so</code> , but is not automatically linked by the C compilation system. Specify <code>-lsecdb</code> on the <code>cc</code> command line to link with this library. See <code>libsecdb(3LIB)</code> .
(3SMARTCARD)	These functions constitute the smartcard library, <code>libsmartcard</code> . This library is implemented as a shared object, <code>libsmartcard.so</code> , but is not automatically linked by the C compilation system. Specify <code>-lsmartcard</code> on the <code>cc</code> command line to link with this library. See <code>libsmartcard(3LIB)</code> .
(3SNMP)	These functions constitute the SNMP libraries, <code>libdssagent</code> and <code>libdssasmp</code> . These libraries are implemented as shared objects, <code>libssagent.so</code> and <code>libssasmp.so</code> , respectively, but are not automatically linked by the C compilation system. Specify <code>-lssagent</code> or <code>-lssasmp</code> on the <code>cc</code> command line to link with these libraries. See <code>libssagent(3LIB)</code> and <code>libssasmp(3LIB)</code> .

(3SYSEVENT)	These functions constitute the system event library, <code>libsysevent</code> . This library is implemented as a shared object, <code>libsysevent.so</code> , but is not automatically linked by the C compilation system. Specify <code>-lsysevent</code> on the <code>cc</code> command line to link with this library. See <code>libsysevent(3LIB)</code> .
(3TNF)	These functions constitute the TNF libraries, <code>libtnf</code> , <code>libtnfctl</code> , and <code>libtnfprobe</code> . These libraries are implemented as shared objects, <code>libtnf.so</code> , <code>libtnfctl.so</code> , and <code>libtnfprobe.so</code> , respectively, but are not automatically linked by the C compilation system. Specify <code>-ltnf</code> , <code>-ltnfctl</code> , or <code>-ltnfprobe</code> on the <code>cc</code> command line to link with these libraries. See <code>libtnfctl(3TNF)</code> and <code>libtnfctl(3LIB)</code> .
(3VOLMGT)	These functions constitute the volume management library, <code>libvolmgt</code> . This library is implemented as a shared object, <code>libvolmgt.so</code> , and as an archive, <code>libvolmgt.a</code> , but is not automatically linked by the C compilation system. Specify <code>-lvolmgt</code> on the <code>cc</code> command line to link with this library. See <code>libvolmgt(3LIB)</code> .
(3WSREG)	These functions constitute the product install registry library, <code>libwsreg</code> . This library is implemented as a shared object, <code>libwsreg.so</code> , but is not automatically linked by the C compilation system. Specify <code>-lwsreg</code> on the <code>cc</code> command line to link with this library. See <code>libwsreg(3LIB)</code> .

DEFINITIONS

A character is any bit pattern able to fit into a byte on the machine. In some international languages, however, a “character” may require more than one byte, and is represented in multi-bytes.

The null character is a character with value 0, conventionally represented in the C language as `\0`. A character array is a sequence of characters. A null-terminated character array (a *string*) is a sequence of characters, the last of which is the null character. The null string is a character array containing only the terminating null character. A null pointer is the value that is obtained by casting 0 into a pointer. C guarantees that this value will not match that of any legitimate pointer, so many functions that return pointers return `NULL` to indicate an error. The macro `NULL` is defined in `<stdio.h>`. Types of the form `size_t` are defined in the appropriate headers.

MT-Level of Libraries FILES

See `attributes(5)` for descriptions of library MT-Levels.

<code>INCDIR</code>	usually <code>/usr/include</code>
<code>LIBDIR</code>	usually <code>/usr/lib</code> (32-bit) or <code>/usr/lib/sparcv9</code> (64-bit)

Intro(3)

LIBDIR/libc.so

LIBDIR/libc.a

LIBDIR/libgen.a

LIBDIR/libm.a

LIBDIR/libsfm.sa

/usr/lib/libc.so.1

SEE ALSO *ar(1)*, *cc(1B)*, *ld(1)*, *fork(2)*, *intro(3)*, *stdio(3C)*, *attributes(5)*, *standards(5)*

Linker and Libraries Guide

Profiling Tools

ANSI C Programmer's Guide

DIAGNOSTICS For functions that return floating-point values, error handling varies according to compilation mode. Under the *-xt* (default) option to *cc*, these functions return the conventional values 0, \pm HUGE, or NaN when the function is undefined for the given arguments or when the value is not representable. In the *-Xa* and *-Xc* compilation modes, \pm HUGE_VAL is returned instead of \pm HUGE. (HUGE_VAL and HUGE are defined in *math.h* to be infinity and the largest-magnitude single-precision number, respectively.)

NOTES ON MULTITHREADED APPLICATIONS When compiling a multithreaded application, either the `_POSIX_C_SOURCE` or `_POSIX_PTHREAD_SEMANTICS` flag or the *-mt* option must be specified on the command line. This enables special definitions for functions only applicable to multithreaded applications. For POSIX.1c-conforming applications, define the `_POSIX_C_SOURCE` flag to be \geq 199506L:

```
cc [flag...] file... -D_POSIX_C_SOURCE=199506L -pthread
```

For POSIX behavior with the Solaris *fork()* and *fork1()* distinction, compile as follows:

```
cc [flag...] file... -D_POSIX_PTHREAD_SEMANTICS -pthread
```

For Solaris threads behavior, compile as follows:

```
cc - mt [ flag... ] file...
```

When building a singlethreaded application, the above *flag* arguments should be undefined.

Unsafe interfaces should be called only from the main thread to ensure the application's safety.

**REALTIME
APPLICATIONS**

MT-Safe interfaces are denoted in the `ATTRIBUTES` section of the functions and libraries manual pages (see `attributes(5)`). If a manual page does not state explicitly that an interface is MT-Safe, the user should assume that the interface is unsafe.

Be sure to have set the environment variable `LD_BIND_NOW` to a non-null value to enable early binding. Refer to the “When Relocations are Processed” chapter in *Linker and Libraries Guide* for additional information.

NOTES

None of the functions, external variables, or macros should be redefined in the user’s programs. Any other name may be redefined without affecting the behavior of other library functions, but such redefinition may conflict with a declaration in an included header.

The headers in `INCDIR` provide function prototypes (function declarations including the types of arguments) for most of the functions listed in this manual. Function prototypes allow the compiler to check for correct usage of these functions in the user’s program. The `lint` program checker may also be used and will report discrepancies even if the headers are not included with `#include` statements. Definitions for Sections 2, 3C, and 3S are checked automatically. Other definitions can be included by using the `-l` option to `lint`. (For example, `-lm` includes definitions for `libm`.) Use of `lint` is highly recommended. See the `lint` chapter in *Performance Profiling Tools*.

Users should carefully note the difference between `STREAMS` and *stream*. `STREAMS` is a set of kernel mechanisms that support the development of network services and data communication drivers. It is composed of utility routines, kernel facilities, and a set of data structures. A *stream* is a file with its associated buffering. It is declared to be a pointer to a type `FILE` defined in `<stdio.h>`.

In detailed definitions of components, it is sometimes necessary to refer to symbolic names that are implementation-specific, but which are not necessarily expected to be accessible to an application program. Many of these symbolic names describe boundary conditions and system limits.

In this section, for readability, these implementation-specific values are given symbolic names. These names always appear enclosed in curly brackets to distinguish them from symbolic names of other implementation-specific constants that are accessible to application programs by headers. These names are not necessarily accessible to an application program through a header, although they may be defined in the documentation for a particular system.

In general, a portable application program should not refer to these symbolic names in its code. For example, an application program would not be expected to test the length of an argument list given to a routine to determine if it was greater than `{ARG_MAX}`.

The Federated Naming Service based on the X/Open XFN standard might not be supported in a future release of the Solaris operating system.

Intro(3)

Library Interfaces and Headers

acct(3HEAD)

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

```

/*
 * 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 */
};

```

The `ta_cpu`, `ta_kcore`, and `ta_con` members 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)`).

The `ta_kcore` member 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.

The `ta_con` member 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.

aio(3HEAD)

NAME	aio – asynchronous input and output																					
SYNOPSIS	<pre>#include <aio.h></pre>																					
DESCRIPTION	<p>The <code><aio.h></code> header defines the <code>aio_cb</code> structure which includes the following members:</p> <table><tr><td><code>int</code></td><td><code>aio_fildes</code></td><td>file descriptor</td></tr><tr><td><code>off_t</code></td><td><code>aio_offset</code></td><td>file offset</td></tr><tr><td><code>volatile void*</code></td><td><code>aio_buf</code></td><td>location of buffer</td></tr><tr><td><code>size_t</code></td><td><code>aio_nbytes</code></td><td>length of transfer</td></tr><tr><td><code>int</code></td><td><code>aio_reqprio</code></td><td>request priority offset</td></tr><tr><td><code>struct sigevent</code></td><td><code>aio_sigevent</code></td><td>signal number and value</td></tr><tr><td><code>int</code></td><td><code>aio_lio_opcode</code></td><td>operation to be performed</td></tr></table> <p>This header also includes the following constants:</p> <pre>AIO_CANCELED AIO_NOTCANCELED AIO_ALLDONE LIO_WAIT LIO_NOWAIT LIO_READ LIO_WRITE LIO_NOP</pre>	<code>int</code>	<code>aio_fildes</code>	file descriptor	<code>off_t</code>	<code>aio_offset</code>	file offset	<code>volatile void*</code>	<code>aio_buf</code>	location of buffer	<code>size_t</code>	<code>aio_nbytes</code>	length of transfer	<code>int</code>	<code>aio_reqprio</code>	request priority offset	<code>struct sigevent</code>	<code>aio_sigevent</code>	signal number and value	<code>int</code>	<code>aio_lio_opcode</code>	operation to be performed
<code>int</code>	<code>aio_fildes</code>	file descriptor																				
<code>off_t</code>	<code>aio_offset</code>	file offset																				
<code>volatile void*</code>	<code>aio_buf</code>	location of buffer																				
<code>size_t</code>	<code>aio_nbytes</code>	length of transfer																				
<code>int</code>	<code>aio_reqprio</code>	request priority offset																				
<code>struct sigevent</code>	<code>aio_sigevent</code>	signal number and value																				
<code>int</code>	<code>aio_lio_opcode</code>	operation to be performed																				
SEE ALSO	<code>lseek(2)</code> , <code>read(2)</code> , <code>write(2)</code> , <code>fsync(3C)</code>																					

NAME	ar – archive file format
SYNOPSIS	<pre>#include <ar.h></pre>
DESCRIPTION	<p>The archive command <code>ar</code> is used to combine several files into one. Archives are used mainly as libraries to be searched by the link editor <code>ld</code>.</p> <p>Each archive begins with the archive magic string.</p> <pre>#define ARMAG "!<arch>\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_fmag[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 <code>ar</code> 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> <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 <code>ld</code> 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 <code>ar</code>.</p>

ar(3HEAD)

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.

```
                                0      1      2      3
0x01020304                    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: $ar_size - 4 \text{ bytes} * (\text{“the number of symbols”} + 1)$.

As an example, the following symbol table defines 4 symbols. The archive member at file offset 114 defines *name*. The archive member at file offset 122 defines *object*. The archive member at file offset 426 defines *function* and the archive member at file offset 434 defines *name2*.

Example Symbol Table

Offset	+0	+1	+2	+3	
0	4				4 offset entries
4	114				name
8	122				object
12	426				function
16	434				name2
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	2	\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
short-name	short-name/ Not in string table
file_name_sample	/0 Offset 0 in string table
longerfilenameexample	/18 Offset 18 in string table

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

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

dirent(3HEAD)

NAME	dirent – file system independent directory entry
SYNOPSIS	<pre>#include <dirent.h></pre>
DESCRIPTION	<p>Different file system types may have different directory entries. The <code>dirent</code> 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 <code>getdents(2)</code> system call.</p> <p>The <code>dirent</code> 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 <code>d_ino</code> is a number which is unique for each file in the file system. The <code>d_off</code> entry contains a value which is interpretable only by the filesystem that generated it. It may be supplied as an offset to <code>lseek(2)</code> to find the entry following the current one in a directory. The field <code>d_name</code> is the beginning of the character array giving the name of the directory entry. This name is null terminated and may have at most <code>MAXNAMLEN</code> characters. This results in file system independent directory entries being variable length entities. The value of <code>d_reclen</code> 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>
SEE ALSO	<code>getdents(2)</code> , <code>lseek(2)</code>

NAME	fcntl – file control options																																								
SYNOPSIS	#include <fcntl.h>																																								
DESCRIPTION	<p>The <fcntl.h> header defines the following requests and arguments for use by the functions <code>fcntl(2)</code>, <code>open(2)</code>, and <code>openat(2)</code>.</p> <p>Values for <i>cmd</i> used by <code>fcntl()</code> (the following values are unique):</p> <table> <tr> <td><code>F_DUPFD</code></td> <td>Duplicate file descriptor.</td> </tr> <tr> <td><code>F_DUP2FD</code></td> <td>Similar to <code>F_DUPFD</code>, but always returns <i>arg</i>.</td> </tr> <tr> <td><code>F_GETFD</code></td> <td>Get file descriptor flags.</td> </tr> <tr> <td><code>F_SETFD</code></td> <td>Set file descriptor flags.</td> </tr> <tr> <td><code>F_GETFL</code></td> <td>Get file status flags.</td> </tr> <tr> <td><code>F_SETFL</code></td> <td>Set file status flags.</td> </tr> <tr> <td><code>F_GETOWN</code></td> <td>Get process or process group ID to receive SIGURG signals.</td> </tr> <tr> <td><code>F_SETOWN</code></td> <td>Set process or process group ID to receive SIGURG signals.</td> </tr> <tr> <td><code>F_FREESP</code></td> <td>Free storage space associated with a section of the ordinary file <i>files</i>.</td> </tr> <tr> <td><code>F_GETLK</code></td> <td>Get record locking information.</td> </tr> <tr> <td><code>F_GETLK64</code></td> <td>Equivalent to <code>F_GETLK</code>, but takes a <code>struct flock64</code> argument rather than a <code>struct flock</code> argument.</td> </tr> <tr> <td><code>F_SETLK</code></td> <td>Set record locking information.</td> </tr> <tr> <td><code>F_SETLK64</code></td> <td>Equivalent to <code>F_SETLK</code>, but takes a <code>struct flock64</code> argument rather than a <code>struct flock</code> argument.</td> </tr> <tr> <td><code>F_SETLKW</code></td> <td>Set record locking information; wait if blocked.</td> </tr> <tr> <td><code>F_SETLKW64</code></td> <td>Equivalent to <code>F_SETLKW</code>, but takes a <code>struct flock64</code> argument rather than a <code>struct flock</code> argument.</td> </tr> <tr> <td><code>F_SHARE</code></td> <td>Set share reservation.</td> </tr> <tr> <td><code>F_UNSHARE</code></td> <td>Remove share reservation.</td> </tr> </table> <p>File descriptor flags used for <code>fcntl()</code>:</p> <table> <tr> <td><code>FD_CLOEXEC</code></td> <td>Close the file descriptor upon execution of an <code>exec</code> function (see <code>exec(2)</code>).</td> </tr> </table> <p>Values for <i>l_type</i> used for record locking with <code>fcntl()</code> (the following values are unique):</p> <table> <tr> <td><code>F_RDLCK</code></td> <td>Shared or read lock.</td> </tr> <tr> <td><code>F_UNLCK</code></td> <td>Unlock.</td> </tr> </table>	<code>F_DUPFD</code>	Duplicate file descriptor.	<code>F_DUP2FD</code>	Similar to <code>F_DUPFD</code> , but always returns <i>arg</i> .	<code>F_GETFD</code>	Get file descriptor flags.	<code>F_SETFD</code>	Set file descriptor flags.	<code>F_GETFL</code>	Get file status flags.	<code>F_SETFL</code>	Set file status flags.	<code>F_GETOWN</code>	Get process or process group ID to receive SIGURG signals.	<code>F_SETOWN</code>	Set process or process group ID to receive SIGURG signals.	<code>F_FREESP</code>	Free storage space associated with a section of the ordinary file <i>files</i> .	<code>F_GETLK</code>	Get record locking information.	<code>F_GETLK64</code>	Equivalent to <code>F_GETLK</code> , but takes a <code>struct flock64</code> argument rather than a <code>struct flock</code> argument.	<code>F_SETLK</code>	Set record locking information.	<code>F_SETLK64</code>	Equivalent to <code>F_SETLK</code> , but takes a <code>struct flock64</code> argument rather than a <code>struct flock</code> argument.	<code>F_SETLKW</code>	Set record locking information; wait if blocked.	<code>F_SETLKW64</code>	Equivalent to <code>F_SETLKW</code> , but takes a <code>struct flock64</code> argument rather than a <code>struct flock</code> argument.	<code>F_SHARE</code>	Set share reservation.	<code>F_UNSHARE</code>	Remove share reservation.	<code>FD_CLOEXEC</code>	Close the file descriptor upon execution of an <code>exec</code> function (see <code>exec(2)</code>).	<code>F_RDLCK</code>	Shared or read lock.	<code>F_UNLCK</code>	Unlock.
<code>F_DUPFD</code>	Duplicate file descriptor.																																								
<code>F_DUP2FD</code>	Similar to <code>F_DUPFD</code> , but always returns <i>arg</i> .																																								
<code>F_GETFD</code>	Get file descriptor flags.																																								
<code>F_SETFD</code>	Set file descriptor flags.																																								
<code>F_GETFL</code>	Get file status flags.																																								
<code>F_SETFL</code>	Set file status flags.																																								
<code>F_GETOWN</code>	Get process or process group ID to receive SIGURG signals.																																								
<code>F_SETOWN</code>	Set process or process group ID to receive SIGURG signals.																																								
<code>F_FREESP</code>	Free storage space associated with a section of the ordinary file <i>files</i> .																																								
<code>F_GETLK</code>	Get record locking information.																																								
<code>F_GETLK64</code>	Equivalent to <code>F_GETLK</code> , but takes a <code>struct flock64</code> argument rather than a <code>struct flock</code> argument.																																								
<code>F_SETLK</code>	Set record locking information.																																								
<code>F_SETLK64</code>	Equivalent to <code>F_SETLK</code> , but takes a <code>struct flock64</code> argument rather than a <code>struct flock</code> argument.																																								
<code>F_SETLKW</code>	Set record locking information; wait if blocked.																																								
<code>F_SETLKW64</code>	Equivalent to <code>F_SETLKW</code> , but takes a <code>struct flock64</code> argument rather than a <code>struct flock</code> argument.																																								
<code>F_SHARE</code>	Set share reservation.																																								
<code>F_UNSHARE</code>	Remove share reservation.																																								
<code>FD_CLOEXEC</code>	Close the file descriptor upon execution of an <code>exec</code> function (see <code>exec(2)</code>).																																								
<code>F_RDLCK</code>	Shared or read lock.																																								
<code>F_UNLCK</code>	Unlock.																																								

fcntl(3HEAD)

F_WRLCK Exclusive or write lock.

Values for `f_access` used for share reservations with `fcntl()` (the following values are unique):

F_RDACC Read-only share reservation.

F_WRACC Write-only share reservation.

F_RWACC Read and write share reservation.

Values for `f_deny` used for share reservations with `fcntl()` (the following values are unique):

F_COMPAT Compatibility mode share reservation.

F_RDDNY Deny other read access share reservations.

F_WRDNY Deny other write access share reservations.

F_RWDNY Deny other read or write access share reservations.

F_NODNY Do not deny other read or write access share reservations.

File creation and assignment flags are used in the *oflag* argument by `open()` and `openat()`. All of these values are bitwise distinct:

O_CREAT Create file if it does not exist.

O_EXCL Exclusive use flag.

O_NOCTTY Do not assign controlling tty.

O_TRUNC Truncate flag.

O_XATTR When opening a file, this flag affects the way in which relative paths are resolved by `open()` and `openat()`. With this flag set, the *path* argument is resolved as an extended attribute reference on either the current working directory (if `open()`) or of the file referenced by the file descriptor argument of `openat()`.

File status flags used for `fcntl()`, `open()`, and `open()`:

O_APPEND Set append mode.

O_NDELAY Non-blocking mode.

O_NONBLOCK Non-blocking mode (POSIX; see `standards(5)`).

O_DSYNC Write I/O operations on the file descriptor complete as defined by synchronized I/O data integrity completion.

O_RSYNC Read I/O operations on the file descriptor complete at the same level of integrity as specified by the `O_DSYNC` and `O_SYNC` flags. If both `O_DSYNC` and `O_RSYNC` are set in *oflag*, all I/O operations on the file descriptor complete as defined by

synchronized I/O data integrity completion. If both `O_SYNC` and `O_RSYNC` are set in *oflag*, all I/O operations on the file descriptor complete as defined by synchronized I/O file integrity completion.

`O_SYNC` When opening a regular file, this flag affects subsequent writes. If set, each `write(2)` will wait for both the file data and file status to be physically updated. Write I/O operations on the file descriptor complete as defined by synchronized I/O file integrity completion.

Mask for use with file access modes:

`O_ACCMODE` Mask for file access modes.

File access modes used for `fcntl()`, `open()`, and `openat()`:

`O_RDONLY` Open for reading only.

`O_RDWR` Open for reading and writing.

`O_WRONLY` Open for writing only.

The following constants are used by system calls capable of resolving paths relative to a provided open file descriptor:

`AT_FDCWD` Special value to pass in place of a file descriptor to inform the called routine that relative path arguments should be resolved from the current working directory.

`AT_SYMLINK_NOFOLLOW` Flag passed to `fstatat(2)` and `fchownat(2)` to change the behavior of these functions when they are given a file as an argument that is a symbolic link. In this case the functions operate on the symbolic link file rather than the file the link references.

`AT_REMOVEDIR` Flag passed to `unlinkat(2)` to tell it to assume that its path argument refers to a directory and to attempt to remove this directory.

The `flock` structure describes a file lock. It includes the following members:

```
short  l_type; /* Type of lock */
short  l_whence; /* Flag for starting offset */
off_t  l_start; /* Relative offset in bytes */
off_t  l_len; /* Size; if 0 then until EOF */
long   l_sysid; /* Returned with F_GETLK */
pid_t  l_pid; /* Returned with F_GETLK */
```

The structure `fshare` describes a file share reservation. It includes the following members:

```
short  f_access; /* Type of reservation */
short  f_deny; /* Type of reservations to deny */
long   f_id; /* Process unique identifier */
```

fcntl(3HEAD)

SEE ALSO `creat(2)`, `exec(2)`, `fcntl(2)`, `open(2)`, `fdatasync(3RT)`, `fsync(3C)`, `fsattr(5)`, `standards(5)`

NOTES Data is successfully transferred for a write operation to a regular file when the system ensures that all data written is readable on any subsequent open of the file (even one that follows a system or power failure) in the absence of a failure of the physical storage medium.

Data is successfully transferred for a read operation when an image of the data on the physical storage medium is available to the requesting process.

Synchronized I/O data integrity completion (see `fdatasync(3RT)`):

- For reads, the operation has been completed or diagnosed if unsuccessful. The read is complete only when an image of the data has been successfully transferred to the requesting process. If there were any pending write requests affecting the data to be read at the time that the synchronized read operation was requested, these write requests will be successfully transferred prior to reading the data.
- For writes, the operation has been completed or diagnosed if unsuccessful. The write is complete only when the data specified in the write request is successfully transferred, and all file system information required to retrieve the data is successfully transferred.

File attributes that are not necessary for data retrieval (access time, modification time, status change time) need not be successfully transferred prior to returning to the calling process.

Synchronized I/O file integrity completion (see `fsync(3C)`):

- Identical to a synchronized I/O data integrity completion with the addition that all file attributes relative to the I/O operation (including access time, modification time, status change time) will be successfully transferred prior to returning to the calling process.

NAME	floatingpoint – IEEE floating point definitions	
SYNOPSIS	#include <floatingpoint.h>	
DESCRIPTION	This file defines constants, types, and functions used to implement standard floating point according to ANSI/IEEE Std 754-1985. The functions are implemented in <code>libc</code> . The included header file <code><sys/ieee_fp.h></code> defines certain types of interest to the kernel.	
IEEE Rounding Modes	<code>fp_direction_type</code>	The type of the IEEE rounding direction mode. Note: the order of enumeration varies according to hardware.
	<code>fp_precision_type</code>	The type of the IEEE rounding precision mode, which only applies on systems that support extended precision such as machines based on the Intel 80387 FPU or the 80486. SIGFPE handling:
	<code>sigfpe_code_type</code>	The type of a SIGFPE code.
	<code>sigfpe_handler_type</code>	The type of a user-definable SIGFPE exception handler called to handle a particular SIGFPE code.
	<code>SIGFPE_DEFAULT</code>	A macro indicating the default SIGFPE exception handling, namely to perform the exception handling specified by the user, if any, and otherwise to dump core using <code>abort(3C)</code> .
	<code>SIGFPE_IGNORE</code>	A macro indicating an alternate SIGFPE exception handling, namely to ignore and continue execution.
	<code>SIGFPE_ABORT</code>	A macro indicating an alternate SIGFPE exception handling, namely to abort with a core dump.
IEEE Exception Handling	<code>N_IEEE_EXCEPTION</code>	The number of distinct IEEE floating-point exceptions.
	<code>fp_exception_type</code>	The type of the <code>N_IEEE_EXCEPTION</code> exceptions. Each exception is given a bit number.
	<code>fp_exception_field_type</code>	The type intended to hold at least <code>N_IEEE_EXCEPTION</code> bits corresponding to the IEEE exceptions numbered by <code>fp_exception_type</code> . Thus <code>fp_inexact</code> corresponds to the least significant bit and <code>fp_invalid</code> to the fifth least significant bit. Note: some operations may set more

floatingpoint(3HEAD)

		than one exception.
IEEE Formats and Classification	single; extended; quadruple	Definitions of IEEE formats.
	fp_class_type	An enumeration of the various classes of IEEE values and symbols.
IEEE Base Conversion	The functions described under <code>floating_to_decimal(3C)</code> and <code>decimal_to_floating(3C)</code> satisfy not only the IEEE Standard, but also the stricter requirements of correct rounding for all arguments.	
	DECIMAL_STRING_LENGTH	The length of a <code>decimal_string</code> .
	decimal_string	The digit buffer in a <code>decimal_record</code> .
	decimal_record	The canonical form for representing an unpacked decimal floating-point number.
	decimal_form	The type used to specify fixed or floating binary to decimal conversion.
	decimal_mode	A struct that contains specifications for conversion between binary and decimal.
	decimal_string_form	An enumeration of possible valid character strings representing floating-point numbers, infinities, or NaNs.
	FILES	/usr/include/sys/ieee.h
SEE ALSO	abort(3C), decimal_to_floating(3C), econvert(3C), floating_to_decimal(3C), sigfpe(3C), string_to_decimal(3C), strtod(3C)	

NAME	in – Internet Protocol family
SYNOPSIS	<code>#include <netinet/in.h></code>
DESCRIPTION	<p>The <code><netinet/in.h></code> header defines the following types through <code>typedef</code>:</p> <p><code>in_port_t</code> An unsigned integral type of exactly 16 bits.</p> <p><code>in_addr_t</code> An unsigned integral type of exactly 32 bits. The <code><netinet/in.h></code> header defines the <code>in_addr</code> structure that includes the following member:</p> <p>The <code><netinet/in.h></code> header defines the <code>in_addr</code> structure that includes the following member:</p> <pre>in_addr_t s_addr</pre> <p>The <code><netinet/in.h></code> header defines the type <code>sa_family_t</code> as described in <code>socket(3HEAD)</code>.</p> <p>The <code><netinet/in.h></code> header defines the following macros for use as values of the <i>level</i> argument of <code>getsockopt()</code> and <code>setsockopt()</code>:</p> <p><code>IPPROTO_IP</code> Dummy for IP</p> <p><code>IPPROTO_ICMP</code> Control message protocol</p> <p><code>IPPROTO_TCP</code> TCP</p> <p><code>IPPROTO_UDP</code> User datagram protocol The <code><netinet/in.h></code> header defines the following macros for use as destination addresses for <code>connect()</code>, <code>sendmsg()</code>, and <code>sendto()</code>:</p> <p><code>INADDR_ANY</code> Local host address</p> <p><code>INADDR_BROADCAST</code> Broadcast address</p> <p>The <code><netinet/in.h></code> header defines the <code>sockaddr_in</code> structure that is used to store addresses for the Internet protocol family. Values of this type must be cast to <code>struct sockaddr</code> for use with the <code>socket</code> interfaces.</p>
Default	<p>For applications that do not require standard-conforming behavior (those that use the <code>socket</code> interfaces described in section (3SOCKET) of the reference manual; see <code>Intro(3)</code> and <code>standards(5)</code>), the <code><netinet/in.h></code> header defines the <code>sockaddr_in</code> structure that includes the following members:</p> <pre>sa_family_t sin_family in_port_t sin_port struct in_addr sin_addr char sin_zero[8]</pre>

in(3HEAD)

Standard conforming

For applications that require standard-conforming behavior (those that use the socket interfaces described in section (3XNET) of the reference manual; see Intro(3) and standards(5)), the `<netinet/in.h>` header defines the `sockaddr_in` structure that includes the following members:

```
sa_family_t    sin_family
in_port_t      sin_port
struct in_addr sin_addr
unsigned char  sin_zero[8]
```

SEE ALSO

Intro(3), connect(3SOCKET), connect(3XNET), getsockopt(3SOCKET), getsockopt(3XNET), sendmsg(3SOCKET), sendmsg(3XNET), sendto(3SOCKET), sendto(3XNET), setsockopt(3SOCKET), setsockopt(3XNET), socket(3HEAD), standards(5)

NAME	inet – definitions for internet operations
SYNOPSIS	<pre>#include <arpa/inet.h></pre>
DESCRIPTION	<p>The <code><arpa/inet.h></code> header defines the type <code>in_port_t</code>, the type <code>in_addr_t</code>, and the <code>in_addr</code> structure, as described in <code>in(3HEAD)</code>.</p> <p>Inclusion of the <code><arpa/inet.h></code> header may also make visible all symbols from <code>in(3HEAD)</code>.</p> <p>The following are declared as functions, and may also be defined as macros:</p> <pre>in_addr_t inet_addr(const char *); in_addr_t inet_lnaof(struct in_addr); struct in_addr inet_makeaddr(in_addr_t, in_addr_t); in_addr_t inet_netof(struct in_addr); in_addr_t inet_network(const char *); char *inet_ntoa(struct in_addr);</pre>
Default	<p>For applications that do not require standard-conforming behavior (those that use the socket interfaces described in section 3N of the reference manual; see <code>Intro(3)</code> and <code>standards(5)</code>), the following may be declared as functions, or defined as macros, or both:</p> <pre>uint32_t htonl(uint32_t); uint16_t htons(uint16_t); uint32_t ntohl(uint32_t); uint16_t ntohs(uint16_t);</pre>
Standard conforming	<p>For applications that require standard-conforming behavior (those that use the socket interfaces described in section 3XN of the reference manual; see <code>Intro(3)</code> and <code>standards(5)</code>), the following may be declared as functions, or defined as macros, or both:</p> <pre>in_addr_t htonl(in_addr_t); in_port_t htons(in_port_t); in_addr_t ntohl(in_addr_t); in_port_t ntohs(in_port_t);</pre>
SEE ALSO	<code>Intro(3)</code> , <code>htonl(3SOCKET)</code> , <code>htonl(3XNET)</code> , <code>inet_addr(3SOCKET)</code> , <code>inet_addr(3XNET)</code> , <code>in(3HEAD)</code> , <code>standards(5)</code>

langinfo(3HEAD)

NAME	langinfo – language information constants
SYNOPSIS	#include <langinfo.h>
DESCRIPTION	<p>The <langinfo.h> header contains the constants used to identify items of langinfo data (see nl_langinfo(3C)). The type of the constant, nl_item, is defined as described in <nl_types.h>.</p> <p>The following constants are defined. The entries under Category indicate in which setlocale(3C) category each item is defined.</p>

Constant	Category	Meaning
CODESET	LC_CTYPE	codeset name
D_T_FMT	LC_TIME	string for formatting date and time
D_FMT	LC_TIME	date format string
T_FMT	LC_TIME	time format string
T_FMT_AMPM	LC_TIME	a.m. or p.m. time format string
AM_STR	LC_TIME	ante-meridiem affix
PM_STR	LC_TIME	post-meridiem affix
DAY_1	LC_TIME	name of the first day of the week (for example, Sunday)
DAY_2	LC_TIME	name of the second day of the week (for example, Monday)
DAY_3	LC_TIME	name of the third day of the week (for example, Tuesday)
DAY_4	LC_TIME	name of the fourth day of the week (for example, Wednesday)
DAY_5	LC_TIME	name of the fifth day of the week (for example, Thursday)
DAY_6	LC_TIME	name of the sixth day of the week (for example, Friday)
DAY_7	LC_TIME	name of the seventh day of the week (for example, Saturday)
ABDAY_1	LC_TIME	abbreviated name of the first day of the week
ABDAY_2	LC_TIME	abbreviated name of the second day of the week

Constant	Category	Meaning
ABDAY_3	LC_TIME	abbreviated name of the third day of the week
ABDAY_4	LC_TIME	abbreviated name of the fourth day of the week
ABDAY_5	LC_TIME	abbreviated name of the fifth day of the week
ABDAY_6	LC_TIME	abbreviated name of the seventh day of the week
ABDAY_7	LC_TIME	abbreviated name of the seventh day of the week
MON_1	LC_TIME	name of the first month of the year
MON_2	LC_TIME	name of the second month
MON_3	LC_TIME	name of the third month
MON_4	LC_TIME	name of the fourth month
MON_5	LC_TIME	name of the fifth month
MON_6	LC_TIME	name of the sixth month
MON_7	LC_TIME	name of the seventh month
MON_8	LC_TIME	name of the eighth month
MON_9	LC_TIME	name of the ninth month
MON_10	LC_TIME	name of the tenth month
MON_11	LC_TIME	name of the eleventh month
MON_12	LC_TIME	name of the twelfth month
ABMON_1	LC_TIME	abbreviated name of the first month
ABMON_2	LC_TIME	abbreviated name of the second month
ABMON_3	LC_TIME	abbreviated name of the third month
ABMON_4	LC_TIME	abbreviated name of the fourth month
ABMON_5	LC_TIME	abbreviated name of the fifth month
ABMON_6	LC_TIME	abbreviated name of the sixth month
ABMON_7	LC_TIME	abbreviated name of the seventh month
ABMON_8	LC_TIME	abbreviated name of the eighth month
ABMON_9	LC_TIME	abbreviated name of the ninth month

langinfo(3HEAD)

Constant	Category	Meaning
ABMON_10	LC_TIME	abbreviated name of the tenth month
ABMON_11	LC_TIME	abbreviated name of the eleventh month
ABMON_12	LC_TIME	abbreviated name of the twelfth month
ERA	LC_TIME	era description segments
ERA_D_FMT	LC_TIME	era date format string
ERA_D_T_FMT	LC_TIME	era date and time format string
ERA_T_FMT	LC_TIME	era time format string
ALT_DIGITS	LC_TIME	alternative symbols for digits
RADIXCHAR	LC_NUMERIC	radix character
THOUSEP	LC_NUMERIC	separator for thousands
YESEXPR	LC_MESSAGES	affirmative response expression
NOEXPR	LC_MESSAGES	negative response expression
YESSTR	LC_MESSAGES	affirmative response for yes/no queries
NOSTR	LC_MESSAGES	negative response ro yes/no queries
CRNCYSTR	LC_MONETARY	local currency symbol, preceded by '-' if the symbol should appear before the value, '+' if the symbol should appear after the value, or '.' if the symbol should replace the radix character

If the locale's values for `p_cs_precedes` and `n_cs_precedes` do not match, the value of `nl_langinfo(CRNCYSTR)` is unspecified.

The `<langinfo.h>` header declares the following as a function:

```
char *nl_langinfo(nl_item);
```

Inclusion of `<langinfo.h>` header may also make visible all symbols from `<nl_types.h>`.

USAGE Wherever possible, users are advised to use functions compatible with those in the ISO C standard to access items of `langinfo` data. In particular, the `strptime(3C)` function should be used to access date and time information defined in category `LC_TIME`. The `localeconv(3C)` function should be used to access information corresponding to `RADIXCHAR`, `THOUSEP`, and `CRNCYSTR`.

SEE ALSO `mkmsgs(1)`, `localeconv(3C)`, `nl_langinfo(3C)`, `nl_types(3HEAD)`, `setlocale(3C)`, `strptime(3C)`, `standards(5)`

NAME	libadm – general administrative library							
SYNOPSIS	cc [<i>flag</i> ...] <i>file</i> ... -ladm [<i>library</i> ...]							
DESCRIPTION	Functions in this library provide device management, VTOC handling, regular expressions, and packaging routines.							
INTERFACES	The shared object <code>libadm.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.							
	advance	circf						
	compile	devattr						
	devfree	devreserv						
	get_ABI_namelngh	getdev						
	getdgrp	getvol						
	listdev	listdgrp						
	loc1	loc2						
	locs	nbra						
	pkgdir	pkginfo						
	pkgnmchk	pkgparam						
	read_vtoc	reservdev						
	sed	set_ABI_namelngh						
	step	write_vtoc						
FILES	/usr/lib/libadm.a	archive library						
	/usr/lib/libadm.so.1	shared object						
	/usr/lib/64/libadm.so.1	64-bit shared object						
ATTRIBUTES	See <code>attributes(5)</code> for descriptions of the following attributes:							
	<table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>Availability</td> <td>SUNWcsl, SUNWarc (32-bit) SUNWcslx (64-bit)</td> </tr> <tr> <td>MT-Level</td> <td>Unsafe</td> </tr> </tbody> </table>		ATTRIBUTE TYPE	ATTRIBUTE VALUE	Availability	SUNWcsl, SUNWarc (32-bit) SUNWcslx (64-bit)	MT-Level	Unsafe
ATTRIBUTE TYPE	ATTRIBUTE VALUE							
Availability	SUNWcsl, SUNWarc (32-bit) SUNWcslx (64-bit)							
MT-Level	Unsafe							
SEE ALSO	<code>pvs(1)</code> , <code>intro(3)</code> , <code>read_vtoc(3EXT)</code> , <code>attributes(5)</code> , <code>regex(5)</code>							

libaio(3LIB)

NAME libaio – asynchronous I/O library

SYNOPSIS `cc [flag . . .] file . . . -laio [library . . .]`

DESCRIPTION Functions in this library provide routines for asynchronous I/O.

INTERFACES The shared object `libaio.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

<code>aiocancel</code>	<code>aioread</code>	<code>aiowait</code>	<code>aiowrite</code>
<code>assfail</code>	<code>close</code>	<code>fork</code>	<code>sigaction</code>
<code>sigignore</code>	<code>signal</code>	<code>sigset</code>	

The following interfaces are unique to the 32-bit version of this library:

<code>aioread64</code>	<code>aiowrite64</code>
------------------------	-------------------------

FILES `/usr/lib/libaio.so.1` shared object
`/usr/lib/64/libaio.so.1` 64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)
MT-Level	Safe

SEE ALSO `pvs(1)`, `intro(2)`, `intro(3)`, `aiocancel(3AIO)`, `aioread(3AIO)`, `aiowait(3AIO)`, `aiowrite(3AIO)`, `attributes(5)`

NAME libbsdmalloc – memory allocator interface library

SYNOPSIS `cc [flag . . .] file . . . -lbsdmalloc [library . . .]`
`#include <stdlib.h>`

DESCRIPTION Functions in this library provide a collection of `malloc` routines that use BSD semantics.

INTERFACES The shared object `libbsdmalloc.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

`free` `malloc` `realloc`

FILES

- `/usr/lib/libbsdmalloc.a`
archive library
- `/usr/lib/libbsdmalloc.so.1`
shared object
- `/usr/lib/64/libbsdmalloc.so.1`
64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl, SUNWarc (32-bit) SUNWcslx (64-bit)
MT Level	Unsafe

SEE ALSO `pvs(1)`, `intro(3)`, `bsdmalloc(3MALLOC)`, `attributes(5)`

libbasm(3LIB)

NAME	libbasm – basic security library																																																
SYNOPSIS	cc [<i>flag</i> . . .] <i>file</i> . . . -lbasm [<i>library</i> . . .]																																																
DESCRIPTION	Functions in this library provide basic security, library object reuse, and auditing.																																																
INTERFACES	The shared object <code>libbasm.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.																																																
	<table><tr><td><code>au_close</code></td><td><code>au_open</code></td></tr><tr><td><code>au_preselect</code></td><td><code>au_to_arg</code></td></tr><tr><td><code>au_to_arg32</code></td><td><code>au_to_arg64</code></td></tr><tr><td><code>au_to_attr</code></td><td><code>au_to_cmd</code></td></tr><tr><td><code>au_to_data</code></td><td><code>au_to_groups</code></td></tr><tr><td><code>au_to_in_addr</code></td><td><code>au_to_ipc</code></td></tr><tr><td><code>au_to_iport</code></td><td><code>au_to_me</code></td></tr><tr><td><code>au_to_newgroups</code></td><td><code>au_to_opaque</code></td></tr><tr><td><code>au_to_path</code></td><td><code>au_to_process</code></td></tr><tr><td><code>au_to_return</code></td><td><code>au_to_return32</code></td></tr><tr><td><code>au_to_return64</code></td><td><code>au_to_socket</code></td></tr><tr><td><code>au_to_subject</code></td><td><code>au_to_subject_ex</code></td></tr><tr><td><code>au_to_text</code></td><td><code>au_user_mask</code></td></tr><tr><td><code>au_write</code></td><td><code>audit</code></td></tr><tr><td><code>auditon</code></td><td><code>auditsvc</code></td></tr><tr><td><code>endac</code></td><td><code>endauclass</code></td></tr><tr><td><code>endauevent</code></td><td><code>endauser</code></td></tr><tr><td><code>getacdir</code></td><td><code>getacflg</code></td></tr><tr><td><code>getacmin</code></td><td><code>getacna</code></td></tr><tr><td><code>getauclassent</code></td><td><code>getauclassent_r</code></td></tr><tr><td><code>getauclassnam</code></td><td><code>getauclassnam_r</code></td></tr><tr><td><code>getaudit</code></td><td><code>getaudit_addr</code></td></tr><tr><td><code>getauditflagsbin</code></td><td><code>getauditflagschar</code></td></tr><tr><td><code>getauevent</code></td><td><code>getauevent_r</code></td></tr></table>	<code>au_close</code>	<code>au_open</code>	<code>au_preselect</code>	<code>au_to_arg</code>	<code>au_to_arg32</code>	<code>au_to_arg64</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_return32</code>	<code>au_to_return64</code>	<code>au_to_socket</code>	<code>au_to_subject</code>	<code>au_to_subject_ex</code>	<code>au_to_text</code>	<code>au_user_mask</code>	<code>au_write</code>	<code>audit</code>	<code>auditon</code>	<code>auditsvc</code>	<code>endac</code>	<code>endauclass</code>	<code>endauevent</code>	<code>endauser</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>getaudit_addr</code>	<code>getauditflagsbin</code>	<code>getauditflagschar</code>	<code>getauevent</code>	<code>getauevent_r</code>
<code>au_close</code>	<code>au_open</code>																																																
<code>au_preselect</code>	<code>au_to_arg</code>																																																
<code>au_to_arg32</code>	<code>au_to_arg64</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_return32</code>																																																
<code>au_to_return64</code>	<code>au_to_socket</code>																																																
<code>au_to_subject</code>	<code>au_to_subject_ex</code>																																																
<code>au_to_text</code>	<code>au_user_mask</code>																																																
<code>au_write</code>	<code>audit</code>																																																
<code>auditon</code>	<code>auditsvc</code>																																																
<code>endac</code>	<code>endauclass</code>																																																
<code>endauevent</code>	<code>endauser</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>getaudit_addr</code>																																																
<code>getauditflagsbin</code>	<code>getauditflagschar</code>																																																
<code>getauevent</code>	<code>getauevent_r</code>																																																

getauevnam	getauevnam_r
getauevnonam	getauevnum
getauevnum_r	getaudit
getauserent	getauserent_r
getausernam	getausernam_r
getfauditflags	setac
setauclass	setauclassfile
setaudit	setaudit_addr
setauevent	setaueventfile
setaudit	setauser
setauserfile	testac

FILES /usr/lib/libbssm.a archive library
 /usr/lib/libbssm.so.1 shared object
 /usr/lib/64/libbssm.so.1 64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl, SUNWarc (32-bit) SUNWcslx (64-bit)
MT-Level	See individual man page for each function.

SEE ALSO pvs(1), intro(3), attributes(5)

libc(3LIB)

NAME	libc – C library																																												
SYNOPSIS	<code>cc [flag...] file... -lc [library ...]</code>																																												
DESCRIPTION	Functions in this library provide various facilities defined by System V, ANSI C, POSIX, and so on. See <code>standards(5)</code> . In addition, those facilities previously defined in the internationalization and the wide-character libraries are now defined in this library, as are the facilities previously defined in the multithreading libraries, <code>libthread</code> and <code>libpthread</code> .																																												
INTERFACES	The shared object <code>libc.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.																																												
	<table><tr><td><code>__loc1</code></td><td><code>__errno</code></td></tr><tr><td><code>__assert</code></td><td><code>__builtin_alloca</code></td></tr><tr><td><code>__ctype</code></td><td><code>__fbufsize</code></td></tr><tr><td><code>__filbuf</code></td><td><code>__flbf</code></td></tr><tr><td><code>__flsbuf</code></td><td><code>__flt_rounds</code></td></tr><tr><td><code>__fpending</code></td><td><code>__fpurge</code></td></tr><tr><td><code>__freadable</code></td><td><code>__freading</code></td></tr><tr><td><code>__fsetlocking</code></td><td><code>__fwritable</code></td></tr><tr><td><code>__fwriting</code></td><td><code>__huge_val</code></td></tr><tr><td><code>__iob</code></td><td><code>__loc1</code></td></tr><tr><td><code>__major</code></td><td><code>__makedev</code></td></tr><tr><td><code>__minor</code></td><td><code>__nsw_extended_action</code></td></tr><tr><td><code>__nsw_freeconfig</code></td><td><code>__nsw_getconfig</code></td></tr><tr><td><code>__posix_asctime_r</code></td><td><code>__posix_ctime_r</code></td></tr><tr><td><code>__posix_getgrgid_r</code></td><td><code>__posix_getgrnam_r</code></td></tr><tr><td><code>__posix_getlogin_r</code></td><td><code>__posix_getpwnam_r</code></td></tr><tr><td><code>__posix_getpwuid_r</code></td><td><code>__posix_sigwait</code></td></tr><tr><td><code>__posix_ttyname_r</code></td><td><code>__priocntl</code></td></tr><tr><td><code>__priocntlset</code></td><td><code>__pthread_cleanup_pop</code></td></tr><tr><td><code>__pthread_cleanup_push</code></td><td><code>__sysconf_xpg5</code></td></tr><tr><td><code>__xpg4</code></td><td><code>__xpg4_putmsg</code></td></tr><tr><td><code>__xpg4_putpmsg</code></td><td><code>_access</code></td></tr></table>	<code>__loc1</code>	<code>__errno</code>	<code>__assert</code>	<code>__builtin_alloca</code>	<code>__ctype</code>	<code>__fbufsize</code>	<code>__filbuf</code>	<code>__flbf</code>	<code>__flsbuf</code>	<code>__flt_rounds</code>	<code>__fpending</code>	<code>__fpurge</code>	<code>__freadable</code>	<code>__freading</code>	<code>__fsetlocking</code>	<code>__fwritable</code>	<code>__fwriting</code>	<code>__huge_val</code>	<code>__iob</code>	<code>__loc1</code>	<code>__major</code>	<code>__makedev</code>	<code>__minor</code>	<code>__nsw_extended_action</code>	<code>__nsw_freeconfig</code>	<code>__nsw_getconfig</code>	<code>__posix_asctime_r</code>	<code>__posix_ctime_r</code>	<code>__posix_getgrgid_r</code>	<code>__posix_getgrnam_r</code>	<code>__posix_getlogin_r</code>	<code>__posix_getpwnam_r</code>	<code>__posix_getpwuid_r</code>	<code>__posix_sigwait</code>	<code>__posix_ttyname_r</code>	<code>__priocntl</code>	<code>__priocntlset</code>	<code>__pthread_cleanup_pop</code>	<code>__pthread_cleanup_push</code>	<code>__sysconf_xpg5</code>	<code>__xpg4</code>	<code>__xpg4_putmsg</code>	<code>__xpg4_putpmsg</code>	<code>_access</code>
<code>__loc1</code>	<code>__errno</code>																																												
<code>__assert</code>	<code>__builtin_alloca</code>																																												
<code>__ctype</code>	<code>__fbufsize</code>																																												
<code>__filbuf</code>	<code>__flbf</code>																																												
<code>__flsbuf</code>	<code>__flt_rounds</code>																																												
<code>__fpending</code>	<code>__fpurge</code>																																												
<code>__freadable</code>	<code>__freading</code>																																												
<code>__fsetlocking</code>	<code>__fwritable</code>																																												
<code>__fwriting</code>	<code>__huge_val</code>																																												
<code>__iob</code>	<code>__loc1</code>																																												
<code>__major</code>	<code>__makedev</code>																																												
<code>__minor</code>	<code>__nsw_extended_action</code>																																												
<code>__nsw_freeconfig</code>	<code>__nsw_getconfig</code>																																												
<code>__posix_asctime_r</code>	<code>__posix_ctime_r</code>																																												
<code>__posix_getgrgid_r</code>	<code>__posix_getgrnam_r</code>																																												
<code>__posix_getlogin_r</code>	<code>__posix_getpwnam_r</code>																																												
<code>__posix_getpwuid_r</code>	<code>__posix_sigwait</code>																																												
<code>__posix_ttyname_r</code>	<code>__priocntl</code>																																												
<code>__priocntlset</code>	<code>__pthread_cleanup_pop</code>																																												
<code>__pthread_cleanup_push</code>	<code>__sysconf_xpg5</code>																																												
<code>__xpg4</code>	<code>__xpg4_putmsg</code>																																												
<code>__xpg4_putpmsg</code>	<code>_access</code>																																												

_acct	_addseverity
_alarm	_altzone
_assert	_catclose
_catgets	_catopen
_cfgetispeed	_cfgetospeed
_cfsetispeed	_cfsetospeed
_chdir	_chmod
_chown	_chroot
_cleanup	_close
_closedir	_closefrom
_creat	_crypt
_ctermid	_ctype
_cuserid	_daylight
_dup	_dup2
_encrypt	_environ
_execl	_execl
_execlp	_execv
_execve	_execvp
_exit	_exithandle
_fattach	_fchdir
_fchmod	_fchown
_fcntl	_fdetach
_fdopen	_fdwalk
_filbuf	_fileno
_flsbuf	_flushlbf
_fmtmsg	_fork
_fpathconf	_fstat
_fstatvfs	_fsync
_ftok	_getacct
_getcontext	_getcwd

libc(3LIB)

<code>_getdate</code>	<code>_getdate_err</code>
<code>_getdate_err_addr</code>	<code>_getegid</code>
<code>_geteuid</code>	<code>_getexecname</code>
<code>_getgid</code>	<code>_getgrgid</code>
<code>_getgrnam</code>	<code>_getgroups</code>
<code>_getitimer</code>	<code>_getlogin</code>
<code>_getmsg</code>	<code>_getopt</code>
<code>_getpass</code>	<code>_getpgid</code>
<code>_getpgrp</code>	<code>_getpid</code>
<code>_getpmsg</code>	<code>_getppid</code>
<code>_getprojid</code>	<code>_getpwnam</code>
<code>_getpwuid</code>	<code>_getrlimit</code>
<code>_getsid</code>	<code>_getsubopt</code>
<code>_gettaskid</code>	<code>_gettimeofday</code>
<code>_gettxt</code>	<code>_getuid</code>
<code>_getw</code>	<code>_grantpt</code>
<code>_hcreate</code>	<code>_hdestroy</code>
<code>_hsearch</code>	<code>_initgroups</code>
<code>_insque</code>	<code>_iob</code>
<code>_ioctl</code>	<code>_isascii</code>
<code>_isastream</code>	<code>_isatty</code>
<code>_isnan</code>	<code>_isnand</code>
<code>_kill</code>	<code>_lchown</code>
<code>_lfind</code>	<code>_link</code>
<code>_lockf</code>	<code>_longjmp</code>
<code>_lsearch</code>	<code>_lseek</code>
<code>_lstat</code>	<code>_lwp_cond_broadcast</code>
<code>_lwp_cond_reltimedwait</code>	<code>_lwp_cond_signal</code>
<code>_lwp_cond_timedwait</code>	<code>_lwp_cond_wait</code>
<code>_lwp_continue</code>	<code>_lwp_info</code>

_lwp_kill	
_lwp_mutex_lock	_lwp_mutex_trylock
_lwp_mutex_unlock	_lwp_self
_lwp_sema_init	_lwp_sema_post
_lwp_sema_trywait	_lwp_sema_wait
_lwp_suspend	_lwp_suspend2
_makecontext	_memccpy
_memcntl	_mkdir
_mkfifo	_mknod
_mkstemp	_mktemp
_mlock	_mmap
_modf	_monitor
_mount	_mprotect
_msgctl	_msgget
_msgids	_msgrcv
_msgsnap	_msgsnd
_msync	_munlock
_munmap	_mutex_held
_mutex_lock	_nextafter
_nftw	_nice
_nl_langinfo	_nsc_trydoorcall
_nss_XbyY_buf_alloc	_nss_XbyY_buf_free
_nss_netdb_aliases	_ntp_adjtime
_ntp_gettime	_numeric
_open	_opendir
_pathconf	_pause
_pclose	_pipe
_poll	_popen
_profil	_ptrace
_ptsname	_putacct

libc(3LIB)

_putenv	_putmsg
_putpmsg	_putw
_read	_readdir
_readlink	_readv
_remque	_rename
_resolvepath	_rewinddir
_rmdir	_rw_read_held
_rw_write_held	_rwlock_destroy
_sbrk	_scalb
_seekdir	_sema_destroy
_sema_held	_semctl
_semget	_semids
_semop	_sentimedop
_setcontext	_setgid
_setgroups	_setitimer
_setjmp	_setkey
_setpgid	_setpgrp
_setrlimit	_setsid
_settaskid	_setuid
_shmat	_shmctl
_shmdt	_shmget
_shmids	_sibuf
_sigaction	_sigaddset
_sigaltstack	_sigdelset
_sigemptyset	_sigfillset
_sighold	_sigignore
_sigismember	_siglongjmp
_sigpause	_sigpending
_sigprocmask	_sigrelse
_sigsend	_sigsendset

_sigset	_sigsetjmp
_sigsuspend	_sleep
_sobuf	_stack_grow
_stat	_statvfs
_stime	_strdup
_swab	_swapcontext
_symlink	_sync
_sys_buslist	_sys_cldlist
_sys_fpelist	_sys_illlist
_sys_segvlist	_sys_siginfolistp
_sys_siglist	_sys_siglistn
_sys_siglistp	_sys_traplist
_syscall	_sysconf
_sysinfo	_syslog
_tcdrain	_tcflow
_tcflush	_tcgetattr
_tcgetpgrp	_tcgetsid
_tcsendbreak	_tcsetattr
_tcsetpgrp	_tdelete
_tell	_telldir
_tempnam	_tfind
_time	_times
_timezone	_toascii
_tolower	_toupper
_tsearch	_ttyname
_twalk	_tzname
_tzset	_ulimit
_umask	_umount
_umount2	_uname
_unlink	_unlockpt

libc(3LIB)

_utime	_wait
_waitid	_waitpid
_wracct	_write
_writev	_xftw
a64l	abort
abs	access
acct	acl
addsev	addseverity
adjtime	alarm
altzone	ascftime
asctime	asctime_r
atexit	atof
atoi	atol
atoll	attropen
basename	bcmp
bcopy	bindtextdomain
bind_textdomain_codeset	brk
bsd_signal	bsearch
btowc	bzero
calloc	catclose
catgets	catopen
cfgetispeed	cfgetospeed
cfree	cfsetispeed
cfsetospeed	cftime
chdir	chmod
chown	chroot
clearerr	clock
close	closedir
closefrom	closelog
cond_broadcast	cond_destroy

cond_init	cond_signal
cond_timedwait	cond_wait
confstr	creat
crypt	crypt_genhash_impl
crypt_gensalt	crypt_gensalt_impl
csetcol	csetlen
ctermid	ctermid_r
ctime	ctime_r
cuserid	daylight
dcgettext	dcngettext
dbm_clearerr	dbm_close
dbm_delete	dbm_error
dbm_fetch	dbm_firstkey
dbm_nextkey	dbm_open
dbm_store	dcgettext
decimal_to_double	decimal_to_extended
decimal_to_quadruple	decimal_to_single
dgettext	difftime
directio	dirname
div	double_to_decimal
drand48	dup
dup2	econvert
ecvt	encrypt
endgrent	endnetgrent
endpwent	endspent
endusershell	endutent
endutxent	environ
erand48	errno
euccol	euclen
eucscol	execl

libc(3LIB)

execl	execlp
execv	execve
execvp	exit
exportfs	extended_to_decimal
facl	fattach
fchdir	fchmod
fchown	fchownat
fchroot	fclose
fcntl	fconvert
fcvt	fdetach
fdopen	fdopendir
fdwalk	feof
ferror	fflush
ffs	fgetc
fgetgrent	fgetgrent_r
fgetpos	fgetpwent
fgetpwent_r	fgets
fgetspent	fgetspent_r
fgetwc	fgetws
file_to_decimal	fileno
finite	flockfile
fmtmsg	fnmatch
fopen	fork
fork1	fpathconf
fpclass	fpgetmask
fpgetround	fpgetsticky
fprintf	fpsetmask
fpsetround	fpsetsticky
fputc	fputs
fputwc	fputws

fread	free
freopen	frexp
fscanf	fseek
fseeko	fsetpos
fstat	fstatat
fstatfs	fstatvfs
fsync	ftell
ftello	ftime
ftok	ftruncate
ftrylockfile	ftw
func_to_decimal	funlockfile
futimesat	fwide
fwprintf	fwrite
fwscanf	gconvert
gcvt	getacct
getc	getc_unlocked
getchar	getchar_unlocked
getcontext	getcpuid
getcwd	getdate
getdate_err	getdents
getdtablesize	getegid
getenv	geteuid
getexecname	getextmntent
getgid	getgrent
getgrent_r	getgrgid
getgrgid_r	getgrnam
getgrnam_r	getgroups
gethomegroup	gethostid
gethostname	gethrtime
gethrvtime	getitimer

libc(3LIB)

getloadavg	getlogin
getlogin_r	getmntany
getmntent	getmsg
getnetgrent	getnetgrent_r
getopt	getpagesize
getpagesizes	getpass
getpassphrase	getpflags
getpgid	getpgrp
getpid	getpmsg
getppid	getppriv
getpriority	getprojid
getpw	getpwent
getpwent_r	getpwnam
getpwnam_r	getpwuid
getpwuid_r	getrctl
getrlimit	getrusage
gets	getsid
getspent	getspent_r
getspnam	getspnam_r
getsubopt	gettaskid
gettext	gettimeofday
gettxt	getuid
getusershell	getustack
getutent	getutid
getutline	getutmp
getutmpx	getutxent
getutxid	getutxline
getvfsany	getvfssent
getvfsfile	getvfsspec
getw	getwc

getwchar	getwd
getwidth	getws
glob	globfree
gmtime	gmtime_r
grantpt	gsignal
hasmntopt	hcreate
hdestroy	hsearch
iconv	iconv_close
iconv_open	index
initgroups	initstate
innetgr	insque
ioctl	isaexec
isalnum	isalpha
isascii	isastream
isatty	iscntrl
isdigit	isenglish
isgraph	isideogram
islower	isnan
isnand	isnanf
isnumber	isphonogram
isprint	ispunct
issetugid	isspace
isspecial	isupper
iswalnum	iswalpha
iswcntrl	iswctype
iswdigit	iswgraph
iswlower	iswprint
iswpunct	iswspace
iswupper	iswxdigit
isxdigit	jrnd48

libc(3LIB)

kill	killpg
l64a	labs
ladd	lchown
lckpwn	lcong48
ldexp	ldivide
lexp10	lfind
lfmt	lgrp_home
link	
llabs	lldiv
llog10	llseek
lltostr	localeconv
localtime	localtime_r
lockf	logb
lone	longjmp
lrnd48	lsearch
lseek	lshifl
lstat	lsub
lten	lzero
madvise	makecontext
makeutx	malloc
mblen	mbrlen
mbrtowc	mbsinit
mbsrtowcs	mbstowcs
mbtowc	memalign
memccpy	memchr
memcmp	memcntl
memcpy	meminfo
memmove	memset
mincore	mkdir
mkfifo	mknod

mkstemp	mktemp
mktime	mlock
mlockall	mmap
modctl	modf
modff	modutx
monitor	mount
mprotect	mrand48
msgctl	msgget
msgids	msgrcv
msgsnap	msgsnd
msync	munlock
munlockall	munmap
mutex_destroy	mutex_init
mutex_lock	mutex_trylock
mutex_unlock	nextafter
nfs_getfh	nftw
ngettext	nice
nl_langinfo	nrnd48
nss_default_finders	nss_delete
nss_endent	nss_getent
nss_search	nss_setent
ntp_adjtime	ntp_gettime
open	openat
opendir	openlog
optarg	opterr
optind	optopt
p_online	pathconf
pause	pclose
pcsample	perror
pfmt	pipe

libc(3LIB)

plock	poll
popen	pread
printf	printstack
priv_addset	priv_allocset
priv_copyset	priv_delset
priv_emptyset	priv_fillset
priv_freerset	priv_getbyname
priv_getbynum	priv_getsetbyname
priv_getsetbynum	priv_intersect
priv_inverse	priv_isset
priv_isequalset	priv_isfullset
priv_ismember	priv_issubset
priv_set	priv_set_to_str
priv_str_to_set	priv_union
processor_bind	processor_info
profil	pset_assign
pset_bind	pset_create
pset_destroy	pset_getattr
pset_getloadavg	pset_info
pset_list	pset_setattr
psiginfo	psignal
pthread_atfork	pthread_attr_destroy
pthread_attr_getdetachstate	pthread_attr_getguardsize
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_setguardsize	pthread_attr_setinheritsched
pthread_attr_setschedparam	pthread_attr_setschedpolicy
pthread_attr_setscope	pthread_attr_setstackaddr

pthread_attr_setstacksize	pthread_cancel
pthread_cond_broadcast	pthread_cond_destroy
pthread_cond_init	pthread_cond_reltimedwait_np
pthread_cond_signal	pthread_cond_timedwait
pthread_cond_wait	pthread_condattr_destroy
pthread_condattr_getpshared	pthread_condattr_init
pthread_condattr_setpshared	pthread_create
pthread_detach	pthread_equal
pthread_exit	pthread_getconcurrency
pthread_getschedparam	pthread_getspecific
pthread_join	pthread_key_create
pthread_key_delete	pthread_kill
pthread_mutex_destroy	pthread_mutex_getprioceiling
pthread_mutex_init	pthread_mutex_lock
pthread_mutex_setprioceiling	pthread_mutex_trylock
pthread_mutex_unlock	pthread_mutexattr_destroy
pthread_mutexattr_getprioceiling	pthread_mutexattr_getprotocol
pthread_mutexattr_getpshared	pthread_mutexattr_gettype
pthread_mutexattr_init	pthread_mutexattr_setprioceiling
pthread_mutexattr_setprotocol	pthread_mutexattr_setpshared
pthread_mutexattr_settype	pthread_once
pthread_rwlock_destroy	pthread_rwlock_init
pthread_rwlock_rdlock	pthread_rwlock_tryrdlock
pthread_rwlock_trywrlock	pthread_rwlock_unlock
pthread_rwlock_wrlock	pthread_rwlockattr_destroy
pthread_rwlockattr_getpshared	pthread_rwlockattr_init
pthread_rwlockattr_setpshared	pthread_self
pthread_setcancelstate	pthread_setcanceltype
pthread_setconcurrency	pthread_setschedparam
pthread_setspecific	pthread_sigmask

libc(3LIB)

pthread_testcancel	
ptsname	putacct
putc	putc_unlocked
putchar	putchar_unlocked
putenv	putmsg
putpmsg	putpwent
puts	putspt
pututline	pututxline
putw	putwc
putwchar	putws
pwrite	qeconvert
qecvt	qfconvert
qfcvt	qgconvert
qgcvt	qsort
quadruple_to_decimal	raise
rand	rand_r
random	rctl_walk
rctlblk_get_enforced_value	rctlblk_get_firing_time
rctlblk_get_global_action	rctlblk_get_global_flags
rctlblk_get_local_action	rctlblk_get_local_flags
rctlblk_get_privilege	rctlblk_get_recipient_pid
rctlblk_get_value	rctlblk_set_local_action
rctlblk_set_local_flags	rctlblk_set_privilege
rctlblk_set_value	rctlblk_set_value
re_comp	re_exec
read	readdir
readdir_r	readlink
readv	realloc
realpath	reboot
regcmp	regcomp

regerror	regex
regexec	regfree
remove	remque
rename	renameat
resetmnttab	resolvepath
rewind	rewinddir
rindex	rmdir
rw_rdlock	rw_read_held
rw_tryrdlock	rw_trywrlock
rw_unlock	rw_write_held
rw_wrlock	rwlock_destroy
rwlock_init	sbrk
scalb	scanf
seconvert	seed48
seekdir	select
sema_destroy	sema_held
sema_init	sema_post
sema_trywait	sema_wait
semctl	semget
semids	semop
semtimedop	setbuf
setbuffer	setcat
setcontext	setegid
seteuid	setgid
setgrent	setgroups
sethostname	setitimer
setjmp	setkey
setlabel	setlinebuf
setlocale	setlogmask
setnetgrent	setpflags

libc(3LIB)

setpgid	setpgrp
setppriv	setpriority
setpwent	setrctl
setregid	setreuid
setrlimit	setsid
setspent	setstate
settaskid	settimeofday
setuid	setusershell
setustack	setutent
setutxent	setvbuf
sfconvert	sgconvert
shmat	shmctl
shmdt	shmget
shmids	sig2str
sigaction	sigaddset
sigaltstack	sigdelset
sigemptyset	sigfillset
sigfpe	sighold
sigignore	siginterrupt
sigismember	siglongjmp
signal	sigpause
sigpending	sigprocmask
sigrelse	sigsend
sigsendset	sigset
sigsetjmp	sigstack
sigsuspend	sigwait
single_to_decimal	sleep
snprintf	sprintf
rand	rand48
random	sscanf

ssignal	stack_getbounds
stack_inbounds	stack_setbounds
stack_violation	stat
statfs	statvfs
stime	str2sig
strcasecmp	strcat
strchr	strcmp
strcoll	strcpy
strcspn	strdup
strerror	strfmon
strftime	string_to_decimal
strlcat	strncpy
strlen	strncasecmp
strncat	strncmp
strncpy	strpbrk
strptime	strrchr
strsignal	strspn
strstr	strtod
strtok	strtok_r
strtol	strtoll
strtoul	strtoull
strtows	strxfrm
swab	swapcontext
swapctl	swprintf
swscanf	symlink
sync	sync_instruction_memory
syscall	sysconf
sysfs	sysinfo
syslog	system
tcdrain	tcflow

libc(3LIB)

tcflush	tcgetattr
tcgetpgrp	tcgetsid
tcsendbreak	tcsetattr
tcsetpgrp	tdelete
tell	telldir
tempnam	textdomain
tfind	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
time	times
timezone	tmpfile
tmpnam	tmpnam_r
toascii	tolower
toupper	towctrans
tolower	towupper
truncate	tsearch
ttyname	ttyname_r
ttyslot	twalk
tzname	tzset
uadmin	ualarm
ucred_free	ucred_get
ucred_getegid	ucred_geteuid
ucred_getgroups	ucred_getpflags

ucrd_getpid	ucrd_getprivset
ucrd_getrgid	ucrd_getruid
ucrd_getsgid	ucrd_getsuid
ulckpwwdf	ulimit
ulltostr	umask
umount	umount2
uname	ungetc
ungetwc	unlink
unlinkat	unlockpt
unordered	updwtmp
updwtmpx	usleep
ustat	utime
utimes	utmpname
utmpxname	valloc
vfork	vfprintf
vfscanf	vfwprintf
vwscanf	vhangup
vlfmt	vpfmt
vprintf	vscanf
vsprintf	vsprintf
vsscanf	vswprintf
vswscanf	vsyslog
vwprintf	vwscanf
wait	wait3
wait4	waitid
waitpid	walkcontext
watoll	wcrtomb
wscat	wcschr
wscmp	wscoll
wscpy	wscspn

libc(3LIB)

wcsftime	wcslen
wcsncat	wcsncmp
wcsncpy	wcspbrk
wcsrchr	wcsrtombs
wcsspn	wcsstr
wctod	wcstok
wcstol	wcstombs
wcstoul	wcswcs
wcswidth	wcsxfrm
wctob	wctomb
wctrans	wctype
wcwidth	wmemchr
wmemcmp	wmemcpy
wmemmove	wmemset
wordexp	wordfree
wprintf	wracct
write	writev
wscanf	wscasecmp
wscat	wchr
wscmp	wscol
wscoll	wscpy
wscspn	wsdup
wslen	wncasecmp
wsncat	wsncmp
wsncpy	wspbrk
wsprintf	wsrchr
wsscanf	wssp
wstod	wstok
wstol	wstoll
wstostr	wsxfrm

yield

The following interfaces are unique to the 32-bit version of this library:

__div64	__mul64
__posix_readdir_r	__rem64
__udiv64	__urem64
_bufendtab	_creat64
_fstat64	_fstatvfs64
_ftruncate64	_ftw64
_getdents64	_getrlimit64
_lastbuf	_lockf64
_lseek64	_lstat64
_mkstemp64	_mmap64
_nftw64	_open64
_pread64	_pwrite64
_readdir64	_readdir64_r
_s_fcntl	_setrlimit64
_stat64	_statvfs64
_sys_nsig	_tell64
_truncate64	_xftw64
creat64	fgetpos64
fopen64	freopen64
fseeko64	fsetpos64
fstat64	fstatvfs64
ftello64	ftruncate64
ftw64	getdents64
getrlimit64	lockf64
lseek64	lstat64
mkstemp64	mmap64
nftw64	open64

libc(3LIB)

pread64	ptrace
pwrite64	readdir64
readdir64_r	s_fcntl
s_ioctl	select_large_fdset
setrlimit64	stat64
statvfs64	sys_errlist
sys_nerr	tell64
tmpfile64	truncate64

The following interfaces are unique to the 32-bit SPARC version of this library:

.div	.mul
.rem	.stret1
.stret2	.stret4
.stret8	.udiv
.umul	.urem
_Q_add	_Q_cmp
_Q_cmpe	_Q_div
_Q_dtoq	_Q_feq
_Q_fge	_Q_fgt
_Q_fle	_Qflt
_Q_fne	_Q_itoq
_Q_lltoq	_Q_mul
_Q_neg	_Q_qtod
_Q_qtoi	_Q_qtoll
_Q_qtos	_Q_qtou
_Q_qtoull	_Q_sqrt
_Q_stoq	_Q_sub
_Q_ulltoq	_Q_utoq
__dtoll	__dtou
__dtoull	__ftoll

<code>__ftou</code>	<code>__ftoull</code>
<code>__umul64</code>	

The following interfaces are unique to the 32-bit x86 version of this library:

<code>__fpstart</code>	<code>_fp_hw</code>
<code>_fpstart</code>	<code>_fxstat</code>
<code>_lxstat</code>	<code>_nuname</code>
<code>_thr_errno_addr</code>	<code>_xmknod</code>
<code>_xstat</code>	<code>nuname</code>

The following interfaces are unique to the 64-bit SPARC version of this library:

<code>_Qp_add</code>	<code>_Qp_cmp</code>
<code>_Qp_cmpe</code>	<code>_Qp_div</code>
<code>_Qp_dtoq</code>	<code>_Qp_feq</code>
<code>_Qp_fge</code>	<code>_Qp_fgt</code>
<code>_Qp_fle</code>	<code>_Qpflt</code>
<code>_Qp_fne</code>	<code>_Qp_itoq</code>
<code>_Qp_mul</code>	<code>_Qp_neg</code>
<code>_Qp_qtod</code>	<code>_Qp_qtoi</code>
<code>_Qp_qtos</code>	<code>_Qp_qtoui</code>
<code>_Qp_qtoux</code>	<code>_Qp_qtox</code>
<code>_Qp_sqrt</code>	<code>_Qp_stoq</code>
<code>_Qp_sub</code>	<code>_Qp_uitoq</code>
<code>_Qp_uxtoq</code>	<code>_Qp_xtoq</code>
<code>__align_cpy_1</code>	<code>__align_cpy_16</code>
<code>__align_cpy_2</code>	<code>__align_cpy_4</code>
<code>__align_cpy_8</code>	<code>__dtoul</code>
<code>__ftoul</code>	<code>__sparc_utrap_install</code>

FILES /usr/lib/libC.so.1

shared object

libc(3LIB)

/usr/lib/64/libc.so.1

64-bit shared object

ATTRIBUTES

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)
MT-Level	Safe

SEE ALSO

`pvs(1)`, `intro(2)`, `intro(3)`, `attributes(5)`, `lf64(5)`, `standards(5)`

NAME libcfgadm – configuration administration library

SYNOPSIS `cc [flag . . .] file . . . -lcfgadm -ldevinfo -ldl [library . . .]`
`#include <config_admin.h>`

DESCRIPTION Functions in this library provide services for configuration administration.

INTERFACES The shared object `libcfgadm.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

<code>config_ap_id_cmp</code>	<code>config_change_state</code>
<code>config_help</code>	<code>config_list</code>
<code>config_list_ext</code>	<code>config_private_func</code>
<code>config_stat</code>	<code>config_strerror</code>
<code>config_test</code>	<code>config_unload_libs</code>

FILES `/usr/lib/libcfgadm.so.1`
 shared object

`/usr/lib/64/libcfgadm.so.1`
 64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)
MT Level	Mt-Safe

SEE ALSO `pvs(1)`, `cfgadm(1M)`, `intro(3)`, `config_admin(3CFGADM)`, `attributes(5)`

libcpc(3LIB)

NAME	libcpc – CPU performance counter library																												
SYNOPSIS	<code>cc [<i>flag</i> . . .] <i>file</i> . . . -lcpc [<i>library</i> . . .]</code>																												
DESCRIPTION	Functions in this library provide access to CPU performance counters on platforms that contain the appropriate hardware.																												
INTERFACES	The shared object <code>libcpc.so.1</code> provides the public interfaces defined below. See intro(3) for additional information on shared object interfaces.																												
	<table border="0"> <tr> <td><code>cpc_access</code></td> <td><code>cpc_bind_event</code></td> </tr> <tr> <td><code>cpc_count_sys_events</code></td> <td><code>cpc_count_usr_events</code></td> </tr> <tr> <td><code>cpc_event_accum</code></td> <td><code>cpc_event_diff</code></td> </tr> <tr> <td><code>cpc_eventtostr</code></td> <td><code>cpc_getcciname</code></td> </tr> <tr> <td><code>cpc_getcpuref</code></td> <td><code>cpc_getcpuver</code></td> </tr> <tr> <td><code>cpc_getnpic</code></td> <td><code>cpc_getusage</code></td> </tr> <tr> <td><code>cpc_pctx_bind_event</code></td> <td><code>cpc_pctx_invalidate</code></td> </tr> <tr> <td><code>cpc_pctx_rele</code></td> <td><code>cpc_pctx_take_sample</code></td> </tr> <tr> <td><code>cpc_rele</code></td> <td><code>cpc_seterrfn</code></td> </tr> <tr> <td><code>cpc_shared_bind_event</code></td> <td><code>cpc_shared_close</code></td> </tr> <tr> <td><code>cpc_shared_open</code></td> <td><code>cpc_shared_rele</code></td> </tr> <tr> <td><code>cpc_shared_take_sample</code></td> <td><code>cpc_strtoevent</code></td> </tr> <tr> <td><code>cpc_take_sample</code></td> <td><code>cpc_version</code></td> </tr> <tr> <td><code>cpc_walk_names</code></td> <td></td> </tr> </table>	<code>cpc_access</code>	<code>cpc_bind_event</code>	<code>cpc_count_sys_events</code>	<code>cpc_count_usr_events</code>	<code>cpc_event_accum</code>	<code>cpc_event_diff</code>	<code>cpc_eventtostr</code>	<code>cpc_getcciname</code>	<code>cpc_getcpuref</code>	<code>cpc_getcpuver</code>	<code>cpc_getnpic</code>	<code>cpc_getusage</code>	<code>cpc_pctx_bind_event</code>	<code>cpc_pctx_invalidate</code>	<code>cpc_pctx_rele</code>	<code>cpc_pctx_take_sample</code>	<code>cpc_rele</code>	<code>cpc_seterrfn</code>	<code>cpc_shared_bind_event</code>	<code>cpc_shared_close</code>	<code>cpc_shared_open</code>	<code>cpc_shared_rele</code>	<code>cpc_shared_take_sample</code>	<code>cpc_strtoevent</code>	<code>cpc_take_sample</code>	<code>cpc_version</code>	<code>cpc_walk_names</code>	
<code>cpc_access</code>	<code>cpc_bind_event</code>																												
<code>cpc_count_sys_events</code>	<code>cpc_count_usr_events</code>																												
<code>cpc_event_accum</code>	<code>cpc_event_diff</code>																												
<code>cpc_eventtostr</code>	<code>cpc_getcciname</code>																												
<code>cpc_getcpuref</code>	<code>cpc_getcpuver</code>																												
<code>cpc_getnpic</code>	<code>cpc_getusage</code>																												
<code>cpc_pctx_bind_event</code>	<code>cpc_pctx_invalidate</code>																												
<code>cpc_pctx_rele</code>	<code>cpc_pctx_take_sample</code>																												
<code>cpc_rele</code>	<code>cpc_seterrfn</code>																												
<code>cpc_shared_bind_event</code>	<code>cpc_shared_close</code>																												
<code>cpc_shared_open</code>	<code>cpc_shared_rele</code>																												
<code>cpc_shared_take_sample</code>	<code>cpc_strtoevent</code>																												
<code>cpc_take_sample</code>	<code>cpc_version</code>																												
<code>cpc_walk_names</code>																													
FILES	<table border="0"> <tr> <td><code>/usr/lib/libcpc.so.1</code></td> <td>shared object</td> </tr> <tr> <td><code>/usr/lib/64/libcpc.so.1</code></td> <td>64-bit shared object</td> </tr> </table>	<code>/usr/lib/libcpc.so.1</code>	shared object	<code>/usr/lib/64/libcpc.so.1</code>	64-bit shared object																								
<code>/usr/lib/libcpc.so.1</code>	shared object																												
<code>/usr/lib/64/libcpc.so.1</code>	64-bit shared object																												
ATTRIBUTES	See attributes(5) for descriptions of the following attributes:																												
	<table border="1"> <thead> <tr> <th style="text-align: center;">ATTRIBUTE TYPE</th> <th style="text-align: center;">ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>Availability</td> <td>SUNWcpcu (32-bit) SUNWcpcux (64-bit)</td> </tr> <tr> <td>MT-Level</td> <td>Safe</td> </tr> </tbody> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	Availability	SUNWcpcu (32-bit) SUNWcpcux (64-bit)	MT-Level	Safe																						
ATTRIBUTE TYPE	ATTRIBUTE VALUE																												
Availability	SUNWcpcu (32-bit) SUNWcpcux (64-bit)																												
MT-Level	Safe																												
SEE ALSO	cputrack(1) , cpustat(1M) , intro(3) , cpc(3CPC) , attributes(5)																												

libcurses(3LIB)

NAME	libcurses, libtermcap, libtermlib – screen handling and optimization library																																								
SYNOPSIS	cc [<i>flag</i> . . .] <i>file</i> . . . -lcurses [<i>library</i> . . .]																																								
DESCRIPTION	<p>Functions in the <code>libcurses</code> library provide a terminal-independent method of updating character screens with reasonable optimization. The <code>libtermcap</code> and <code>libtermlib</code> libraries are identical to <code>libcurses</code> and are maintained for backward compatibility.</p> <p>See <code>libcurses(3XCURSES)</code> for information about the <code>curses</code> library that conforms to X/Open Curses, Issue 4, Version 2.</p>																																								
INTERFACES	<p>The shared objects <code>libcurses.so.1</code>, <code>libtermcap.so.1</code>, and <code>libtermlib.so.1</code> provide the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.</p> <table><tr><td><code>_getsyx</code></td><td><code>_meta</code></td></tr><tr><td><code>_ring</code></td><td><code>_setecho</code></td></tr><tr><td><code>_setnonl</code></td><td><code>_setqiflush</code></td></tr><tr><td><code>addch</code></td><td><code>addchnstr</code></td></tr><tr><td><code>addchstr</code></td><td><code>addnstr</code></td></tr><tr><td><code>addnwstr</code></td><td><code>addstr</code></td></tr><tr><td><code>addwch</code></td><td><code>addwchnstr</code></td></tr><tr><td><code>addwchstr</code></td><td><code>addwstr</code></td></tr><tr><td><code>attroff</code></td><td><code>attron</code></td></tr><tr><td><code>attrset</code></td><td><code>baudrate</code></td></tr><tr><td><code>beep</code></td><td><code>bkgd</code></td></tr><tr><td><code>bkgdset</code></td><td><code>border</code></td></tr><tr><td><code>box</code></td><td><code>can_change_color</code></td></tr><tr><td><code>cbreak</code></td><td><code>clear</code></td></tr><tr><td><code>clearok</code></td><td><code>clrtoBOT</code></td></tr><tr><td><code>clrtoeol</code></td><td><code>color_content</code></td></tr><tr><td><code>copywin</code></td><td><code>crmode</code></td></tr><tr><td><code>curs_set</code></td><td><code>curserr</code></td></tr><tr><td><code>def_prog_mode</code></td><td><code>def_shell_mode</code></td></tr><tr><td><code>del_curterm</code></td><td><code>delay_output</code></td></tr></table>	<code>_getsyx</code>	<code>_meta</code>	<code>_ring</code>	<code>_setecho</code>	<code>_setnonl</code>	<code>_setqiflush</code>	<code>addch</code>	<code>addchnstr</code>	<code>addchstr</code>	<code>addnstr</code>	<code>addnwstr</code>	<code>addstr</code>	<code>addwch</code>	<code>addwchnstr</code>	<code>addwchstr</code>	<code>addwstr</code>	<code>attroff</code>	<code>attron</code>	<code>attrset</code>	<code>baudrate</code>	<code>beep</code>	<code>bkgd</code>	<code>bkgdset</code>	<code>border</code>	<code>box</code>	<code>can_change_color</code>	<code>cbreak</code>	<code>clear</code>	<code>clearok</code>	<code>clrtoBOT</code>	<code>clrtoeol</code>	<code>color_content</code>	<code>copywin</code>	<code>crmode</code>	<code>curs_set</code>	<code>curserr</code>	<code>def_prog_mode</code>	<code>def_shell_mode</code>	<code>del_curterm</code>	<code>delay_output</code>
<code>_getsyx</code>	<code>_meta</code>																																								
<code>_ring</code>	<code>_setecho</code>																																								
<code>_setnonl</code>	<code>_setqiflush</code>																																								
<code>addch</code>	<code>addchnstr</code>																																								
<code>addchstr</code>	<code>addnstr</code>																																								
<code>addnwstr</code>	<code>addstr</code>																																								
<code>addwch</code>	<code>addwchnstr</code>																																								
<code>addwchstr</code>	<code>addwstr</code>																																								
<code>attroff</code>	<code>attron</code>																																								
<code>attrset</code>	<code>baudrate</code>																																								
<code>beep</code>	<code>bkgd</code>																																								
<code>bkgdset</code>	<code>border</code>																																								
<code>box</code>	<code>can_change_color</code>																																								
<code>cbreak</code>	<code>clear</code>																																								
<code>clearok</code>	<code>clrtoBOT</code>																																								
<code>clrtoeol</code>	<code>color_content</code>																																								
<code>copywin</code>	<code>crmode</code>																																								
<code>curs_set</code>	<code>curserr</code>																																								
<code>def_prog_mode</code>	<code>def_shell_mode</code>																																								
<code>del_curterm</code>	<code>delay_output</code>																																								

delch	deleteln
delkeymap	delscreen
delwin	derwin
doupdate	dupwin
echo	echochar
echowchar	endwin
erase	erasechar
filter	flash
flushinp	getbmap
getch	getmouse
getnwstr	getstr
getwch	getwin
getwstr	halfdelay
has_colors	has_ic
has_il	idcok
idlok	immedok
inch	inchnstr
inchstr	init_color
init_pair	initscr
innstr	innwstr
insch	insdelln
insertln	insnstr
insnwstr	insstr
instr	inswch
inswstr	intrflush
inwch	inwchnstr
inwchstr	inwstr
is_linetouched	is_wintouched
isendwin	keyname
keypad	killchar

libcurses(3LIB)

leaveok	longname
m_addch	m_addstr
m_clear	m_erase
m_initscr	m_move
m_newterm	m_refresh
map_button	meta
mouse_off	mouse_on
mouse_set	move
mvaddch	mvaddchnstr
mvaddchstr	mvaddnstr
mvaddnwstr	mvaddstr
mvaddwch	mvaddwchnstr
mvaddwchstr	mvaddwstr
mvcur	mvdelch
mvderwin	mvgetch
mvgetnwstr	mvgetstr
mvgetwch	mvgetwstr
mvinch	mvinchnstr
mvinchstr	mvinnstr
mvinnwstr	mvinsch
mvinsnstr	mvinsnstr
mvinsstr	mvinstr
mvinswch	mvinswstr
mvinwch	mvinwchnstr
mvinwchstr	mvinwstr
mvprintw	mvscanw
mvwaddch	mvwaddchnstr
mvwaddchstr	mvwaddnstr
mvwaddnwstr	mvwaddstr
mvwaddwch	mvwaddwchnstr

mvwaddwchstr	mvwaddwstr
mvwdelch	mvwgetch
mvwgetnwstr	mvwgetstr
mvwgetwch	mvwgetwstr
mvwin	mvwinch
mvwinchnstr	mvwinchstr
mvwinnstr	mvwinnwstr
mvwinsch	mvwinsnstr
mvwinsnwstr	mvwinsstr
mvwinstr	mvwinswch
mvwinswstr	mvwinwch
mvwinwchnstr	mvwinwchstr
mvwinwstr	mvwprintw
mvwscanw	napms
newkey	newpad
newscreen	newterm
newwin	nl
nocbreak	nocrmode
nodelay	noecho
nonl	noqiflush
noraw	notimeout
overlay	overwrite
pair_content	pechochar
pechowchar	pnoutrefresh
prefresh	printw
putp	putwin
qiflush	raw
redrawwin	refresh
request_mouse_pos	reset_prog_mode
reset_shell_mode	resetty

libcurses(3LIB)

restartterm	riporffline
savetty	scanw
scr_dump	scr_init
scr_restore	scr_set
scrl	scroll
scrollok	set_term
setcurscreen	setscrreg
setsyx	setterm
setupterm	slk_attroff
slk_attron	slk_attrset
slk_clear	slk_init
slk_label	slk_noutrefresh
slk_refresh	slk_restore
slk_set	slk_start
slk_touch	standend
standout	start_color
subpad	subwin
syncok	termattrs
termname	tgetent
tgetflag	tgetnum
tgetstr	tgoto
tigetflag	tigetnum
tigetstr	timeout
touchline	touchwin
tparm	tputs
traceoff	traceon
typeahead	unctrl
ungetch	ungetwch
untouchwin	vidattr
vidputs	vidupdate

vwprintw	vwscanw
waddch	waddchnstr
waddchstr	waddnstr
waddnwstr	waddstr
waddwch	waddwchnstr
waddwchstr	waddwstr
wadjcurpos	wattroff
wattron	wattrset
wbkgd	wbkgdset
wborder	wclear
wclrtoeol	wclrtoeol
wcursyncup	wdelch
wdeleteln	wechochar
wechowchar	werase
wgetch	wgetnstr
wgetnwstr	wgetstr
wgetwch	wgetwstr
whline	winch
winchnstr	winchstr
winnstr	winnwstr
winsch	winsdelln
winsertln	winsnstr
winsnwstr	winsstr
winstr	winswch
winswstr	winwch
winwchnstr	winwchstr
winwstr	wmouse_position
wmove	wmovenextch
wmoveprevch	wnoutrefresh
wprintw	wredrawln

libcurses(3LIB)

wrefresh	wscanw
wscr1	wsetscrreg
wstandend	wstandout
wsyncdown	wsyncup
wtimeout	wtouchln
wvline	

FILES

`/usr/lib/libcurses.a`
archive library

`/usr/lib/libcurses.so.1`
shared object

`/usr/lib/64/libcurses.so.1`
64-bit shared object

`/usr/lib/libtermcap.a`
archive library (hard link to `/usr/lib/libcurses.a`)

`/usr/lib/libtermcap.so.1`
shared object (symbolic link to `/usr/lib/libcurses.so.1`)

`/usr/lib/64/libtermcap.so.1`
64-bit shared object (symbolic link to `/usr/lib/64/libcurses.so.1`)

`/usr/lib/libtermlib.a`
archive library (hard link to `/usr/lib/libcurses.a`)

`/usr/lib/libtermlib.so.1`
shared object (symbolic link to `/usr/lib/libcurses.so.1`)

`/usr/lib/64/libtermlib.so.1`
64-bit shared object (symbolic link to `/usr/lib/64/libcurses.so.1`)

ATTRIBUTES

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl, SUNWarc (32-bit) SUNWcslx (64-bit)
MT-Level	Unsafe

SEE ALSO

`intro(3)`, `curses(3CURSES)`, `libcurses(3LIBUCB)`, `libcurses(3XCURSES)`, `attributes(5)`

NAME	libcurses – SunOS/BSD-compatible screen handling and optimization library			
SYNOPSIS	cc [<i>flag</i> . . .] -I /usr/ucbinclude <i>file</i> . . . -L /usr/libucb \ -R /usr/libucb -lcurses [<i>library</i> . . .]			
DESCRIPTION	Functions in this library provide a terminal-independent method of updating character screens with reasonable optimization, compatible with SunOS/BSD.			
INTERFACES	The shared object <code>libcurses.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.			
	AL	AL_PARM	AM	BC
	BS	BT	CA	CD
	CE	CL	CM	COLS
	CR	CS	DA	DB
	DC	DL	DL_PARM	DM
	DO	DOWN_PARM	Def_term	ED
	EI	EO	GT	HC
	HO	HZ	IC	IM
	IN	IP	K0	K1
	K2	K3	K4	K5
	K6	K7	K8	K9
	KD	KE	KH	KL
	KR	KS	KU	LEFT_PARM
	LINES	LL	MA	MI
	MS	My_term	NC	ND
	NL	NONL	NS	OS
	PC	RC	RIGHT_PARM	SC
	SE	SF	SO	SR
	TA	TE	TI	UC
	UE	UL	UP	UPPERCASE
	UP_PARM	US	VB	VE
	VS	XB	XN	XS
	XT	XX	_echoit	_endwin

libcurses(3LIBUCB)

_pfast	_rawmode	_res_flg	_tty
_tty_ch	_unctrl	box	curscr
delwin	endwin	getcap	gettmode
idlok	initscr	longname	mvcur
mvprintw	mvscanw	mvwin	mvwprintw
mvwscanw	newwin	normtty	overlay
overwrite	printw	scanw	scroll
setterm	stdscr	subwin	touchline
touchwin	ttytype	waddch	waddstr
wclear	wclrtobot	wclrtoeol	wdelch
wdeleteln	werase	wgetch	wgetstr
winsch	winsertln	wmove	wprintw
wrefresh	wscanw	wstandend	wstandout

FILES /usr/libucb/libcurses.a
archive library

/usr/libucb/libcurses.so.1
shared object

/usr/libucb/64/libcurses.so.1
64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT-Level	Unsafe

SEE ALSO intro(3), libcurses(3LIB), libcurses(3XCURSES), attributes(5)

NAME libdbm – database subroutines library

SYNOPSIS `cc [flag . . .] -I /usr/ucbinclude file . . . -L /usr/libucb \`
`-R /usr/libucb -ldb [library . . .]`

DESCRIPTION Functions in this library maintain key/content pairs in a database. The functions will handle very large (a billion blocks) databases and will access a keyed item in one or two file system accesses.

INTERFACES The shared object `libdbm.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

<code>bitno</code>	<code>blkno</code>	<code>calchash</code>	<code>dbmclose</code>	<code>dbminit</code>
<code>dbrdonly</code>	<code>delete</code>	<code>dirbuf</code>	<code>dirf</code>	<code>fetch</code>
<code>firstkey</code>	<code>hashinc</code>	<code>hmask</code>	<code>makdatum</code>	<code>maxbno</code>
<code>nextkey</code>	<code>pagbuf</code>	<code>pagf</code>	<code>store</code>	

FILES `/usr/libucb/libdbm.a`
archive library

`/usr/libucb/libdbm.so.1`
shared object

`/usr/libucb/64/libdbm.so.1`
64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT-Level	Unsafe

SEE ALSO `intro(3)`, `dbm(3UCB)`, `attributes(5)`

libdevid(3LIB)

NAME	libdevid – device ID library								
SYNOPSIS	<pre>cc [<i>flag</i> . . .] <i>file</i> . . . -ldevid [<i>library</i> . . .] #include <devid.h></pre>								
DESCRIPTION	Functions in this library provide unique device IDs for identifying a device, independent of the device name or device number.								
INTERFACES	The shared object <code>libdevid.so.1</code> provides the public interfaces defined below. See intro(3) for additional information on shared object interfaces.								
	<pre>devid_compare devid_deviceid_to_nmlist devid_free devid_free_nmlist devid_get devid_get_minor_name devid_sizeof devid_str_decode devid_str_encode devid_str_free devid_valid</pre>								
FILES	<pre>/usr/lib/libdevid.so.1 shared object. /usr/lib/64/libdevid.so.1 64-bit shared object.</pre>								
ATTRIBUTES	See attributes(5) for description of the following attributes:								
	<table border="1"><thead><tr><th>ATTRIBUTE TYPE</th><th>ATTRIBUTE VALUE</th></tr></thead><tbody><tr><td>Availability</td><td>SUNWcsl (32-bit) SUNWcslx (64-bit)</td></tr><tr><td>MT-Level</td><td>MT-Safe</td></tr><tr><td>Interface Stability</td><td>Stable</td></tr></tbody></table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)	MT-Level	MT-Safe	Interface Stability	Stable
ATTRIBUTE TYPE	ATTRIBUTE VALUE								
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)								
MT-Level	MT-Safe								
Interface Stability	Stable								
SEE ALSO	pvs(1) , intro(3) , attributes(5)								

NAME	libdevinfo – device information library																																														
SYNOPSIS	<code>cc [<i>flag</i> ...] <i>file</i> ... -ldevinfo [<i>library</i> ...]</code>																																														
DESCRIPTION	Functions in this library access information on device configuration.																																														
INTERFACES	The shared object <code>libdevinfo.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.																																														
	<table> <tr><td><code>di_binding_name</code></td><td><code>di_bus_addr</code></td></tr> <tr><td><code>di_child_node</code></td><td><code>di_compatible_names</code></td></tr> <tr><td><code>di_devfs_path</code></td><td><code>di_devfs_path_free</code></td></tr> <tr><td><code>di_devid</code></td><td><code>di_driver_name</code></td></tr> <tr><td><code>di_driver_ops</code></td><td><code>di_drv_first_node</code></td></tr> <tr><td><code>di_drv_next_node</code></td><td><code>di_fini</code></td></tr> <tr><td><code>di_init</code></td><td><code>di_instance</code></td></tr> <tr><td><code>di_minor_class</code></td><td><code>di_minor_devt</code></td></tr> <tr><td><code>di_minor_name</code></td><td><code>di_minor_next</code></td></tr> <tr><td><code>di_minor_nodetype</code></td><td><code>di_minor_spectype</code></td></tr> <tr><td><code>di_minor_type</code></td><td><code>di_node_name</code></td></tr> <tr><td><code>di_nodeid</code></td><td><code>di_parent_node</code></td></tr> <tr><td><code>di_prom_fini</code></td><td><code>di_prom_init</code></td></tr> <tr><td><code>di_prom_prop_data</code></td><td><code>di_prom_prop_lookup_bytes</code></td></tr> <tr><td><code>di_prom_prop_lookup_ints</code></td><td><code>di_prom_prop_lookup_strings</code></td></tr> <tr><td><code>di_prom_prop_name</code></td><td><code>di_prom_prop_next</code></td></tr> <tr><td><code>di_prop_bytes</code></td><td><code>di_prop_devt</code></td></tr> <tr><td><code>di_prop_ints</code></td><td><code>di_prop_lookup_bytes</code></td></tr> <tr><td><code>di_prop_lookup_ints</code></td><td><code>di_prop_lookup_strings</code></td></tr> <tr><td><code>di_prop_name</code></td><td><code>di_prop_next</code></td></tr> <tr><td><code>di_prop_strings</code></td><td><code>di_prop_type</code></td></tr> <tr><td><code>di_sibling_node</code></td><td><code>di_state</code></td></tr> <tr><td><code>di_walk_minor</code></td><td><code>di_walk_node</code></td></tr> </table>	<code>di_binding_name</code>	<code>di_bus_addr</code>	<code>di_child_node</code>	<code>di_compatible_names</code>	<code>di_devfs_path</code>	<code>di_devfs_path_free</code>	<code>di_devid</code>	<code>di_driver_name</code>	<code>di_driver_ops</code>	<code>di_drv_first_node</code>	<code>di_drv_next_node</code>	<code>di_fini</code>	<code>di_init</code>	<code>di_instance</code>	<code>di_minor_class</code>	<code>di_minor_devt</code>	<code>di_minor_name</code>	<code>di_minor_next</code>	<code>di_minor_nodetype</code>	<code>di_minor_spectype</code>	<code>di_minor_type</code>	<code>di_node_name</code>	<code>di_nodeid</code>	<code>di_parent_node</code>	<code>di_prom_fini</code>	<code>di_prom_init</code>	<code>di_prom_prop_data</code>	<code>di_prom_prop_lookup_bytes</code>	<code>di_prom_prop_lookup_ints</code>	<code>di_prom_prop_lookup_strings</code>	<code>di_prom_prop_name</code>	<code>di_prom_prop_next</code>	<code>di_prop_bytes</code>	<code>di_prop_devt</code>	<code>di_prop_ints</code>	<code>di_prop_lookup_bytes</code>	<code>di_prop_lookup_ints</code>	<code>di_prop_lookup_strings</code>	<code>di_prop_name</code>	<code>di_prop_next</code>	<code>di_prop_strings</code>	<code>di_prop_type</code>	<code>di_sibling_node</code>	<code>di_state</code>	<code>di_walk_minor</code>	<code>di_walk_node</code>
<code>di_binding_name</code>	<code>di_bus_addr</code>																																														
<code>di_child_node</code>	<code>di_compatible_names</code>																																														
<code>di_devfs_path</code>	<code>di_devfs_path_free</code>																																														
<code>di_devid</code>	<code>di_driver_name</code>																																														
<code>di_driver_ops</code>	<code>di_drv_first_node</code>																																														
<code>di_drv_next_node</code>	<code>di_fini</code>																																														
<code>di_init</code>	<code>di_instance</code>																																														
<code>di_minor_class</code>	<code>di_minor_devt</code>																																														
<code>di_minor_name</code>	<code>di_minor_next</code>																																														
<code>di_minor_nodetype</code>	<code>di_minor_spectype</code>																																														
<code>di_minor_type</code>	<code>di_node_name</code>																																														
<code>di_nodeid</code>	<code>di_parent_node</code>																																														
<code>di_prom_fini</code>	<code>di_prom_init</code>																																														
<code>di_prom_prop_data</code>	<code>di_prom_prop_lookup_bytes</code>																																														
<code>di_prom_prop_lookup_ints</code>	<code>di_prom_prop_lookup_strings</code>																																														
<code>di_prom_prop_name</code>	<code>di_prom_prop_next</code>																																														
<code>di_prop_bytes</code>	<code>di_prop_devt</code>																																														
<code>di_prop_ints</code>	<code>di_prop_lookup_bytes</code>																																														
<code>di_prop_lookup_ints</code>	<code>di_prop_lookup_strings</code>																																														
<code>di_prop_name</code>	<code>di_prop_next</code>																																														
<code>di_prop_strings</code>	<code>di_prop_type</code>																																														
<code>di_sibling_node</code>	<code>di_state</code>																																														
<code>di_walk_minor</code>	<code>di_walk_node</code>																																														
FILES	<code>usr/lib/libdevinfo.a</code> archive library																																														

libdevinfo(3LIB)

/usr/lib/libdevinfo.so.1
shared object

/usr/lib/64/libdevinfo.so.1
64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl, SUNWstatl (32-bit) SUNWcslx (64-bit)
MT Level	Safe
Interface Stability	Evolving

SEE ALSO `pvs(1)`, `intro(3)`, `libdevinfo(3DEVINFO)`, `attributes(5)`

Writing Device Drivers

NAME libdl – dynamic linking library

SYNOPSIS `cc [flag . . .] file . . . -ldl [library . . .]`

DESCRIPTION Functions in this library provide direct access to the dynamic linking facilities. This library is implemented as a filter on the runtime linker (see `ld.so.1(1)`).

INTERFACES The shared object `libdl.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

dladdr dladdr1 dlclose dlclose
dlerror dlerror dlerror dlerror
dlsym

FILES

<code>/usr/lib/libdl.so.1</code>	shared object
<code>/etc/lib/libdl.so.1</code>	shared object (copy)
<code>/usr/lib/64/libdl.so.1</code>	64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)
MT Level	Safe

SEE ALSO `ld.so.1(1)`, `pvs(1)`, `intro(3)`, `attributes(5)`

libdmi(3LIB)

NAME libdmi – Sun Solstice Enterprise Agent DMI library

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

DESCRIPTION The libdmi 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.

INTERFACES The shared object `libdmi.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

```
dmi_error                      freeDmiString
newDmiAttributeValues          newDmiOctetStringFromString
newDmiString                   printDmiDataUnion
printDmiString
```

FILES `/usr/lib/libdmi.so.1` shared object
`/usr/lib/64/libdmi.so.1` 64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWsadmi (32-bit) SUNWsadmx (64-bit)
MT-Level	Unsafe

SEE ALSO `intro(3)`, `libdmici(3LIB)`, `libdmimi(3LIB)`, `attributes(5)`

NAME libdmici – Sun Solstice Enterprise Agent Component library

SYNOPSIS `cc [flag . . .] file . . . -ldmici -ldmi -lnsl -lrwtool \`
`[library . . .]`

DESCRIPTION The libdmici library provides Component Interface API functions.

INTERFACES The shared object libdmici.so.1 provides the public interfaces defined below. See intro(3) for additional information on shared object interfaces.

ConnectToServer	DisconnectToServer
DmiOriginateEvent	DmiRegisterCi
DmiUnregisterCi	reg_ci_callback
start_svc_run_thread	

FILES /usr/lib/libdmici.so.1 shared object
 /usr/lib/sparcv9/libdmici.so.1 64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWsadmi (32-bit) SUNWsadmx (64-bit)
MT-Level	Unsafe

SEE ALSO intro(3), libdmi(3LIB), attributes(5)

libdmimi(3LIB)

NAME libdmimi – Sun Solstice Enterprise Agent Management library

SYNOPSIS `cc [flag . . .] file . . . -ldmimi -ldmi -lnsl -lrwtool \
[library . . .]`

DESCRIPTION The libdmimi library provides Management Interface API functions.

INTERFACES The shared object `libdmimi.so.1` provides the public interfaces defined below. See [intro\(3\)](#) for additional information on shared object interfaces.

<code>ConnectToServer</code>	<code>DisconnectToServer</code>
<code>DmiAddComponent</code>	<code>DmiAddGroup</code>
<code>DmiAddLanguage</code>	<code>DmiAddRow</code>
<code>DmiDeleteComponent</code>	<code>DmiDeleteGroup</code>
<code>DmiDeleteLanguage</code>	<code>DmiDeleteRow</code>
<code>DmiGetAttribute</code>	<code>DmiGetConfig</code>
<code>DmiGetMultiple</code>	<code>DmiGetVersion</code>
<code>DmiListAttributes</code>	<code>DmiListClassNames</code>
<code>DmiListComponents</code>	<code>DmiListComponentsByClass</code>
<code>DmiListGroup</code>	<code>DmiListLanguages</code>
<code>DmiRegister</code>	<code>DmiSetAttribute</code>
<code>DmiSetConfig</code>	<code>DmiSetMultiple</code>
<code>DmiUnregister</code>	

FILES `/usr/lib/libdmimi.so.1` shared object
`/usr/lib/64/libdmimi.so.1` 64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWsdmi (32-bit) SUNWsdmx (64-bit)
MT-Level	Unsafe

SEE ALSO [intro\(3\)](#), [libdmi\(3LIB\)](#), [attributes\(5\)](#)

NAME libefi – EFI partition table library

SYNOPSIS `cc [flag...] file... -lefi [library...]`

DESCRIPTION The functions in this library manipulate a disk's EFI partition table.

INTERFACES The shared object `libefi.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

<code>efi_alloc_and_init</code>	<code>efi_alloc_and_read</code>
<code>efi_free</code>	<code>efi_write</code>

FILES

<code>/usr/lib/libefi.so.1</code>	shared object
<code>/usr/lib/64/libefi.so.1</code>	64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)
Interface Stability	Evolving
MT-Level	Unsafe

SEE ALSO `intro(3)`, `efi_alloc_and_init(3EXT)`, `attributes(5)`

libelf(3LIB)

NAME	libelf – ELF access library
SYNOPSIS	<pre>cc [<i>flag</i> . . .] <i>file</i> . . . -lelf [<i>library</i> . . .] #include <libelf.h></pre>
DESCRIPTION	Functions in this library provide routines to manipulate ELF (Executable and Linking Format) object files, archive files, and archive members. The header provides type and function declarations for all library services.
INTERFACES	The shared object <code>libelf.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.
	<pre>elf32_checksum elf32_fsize elf32_getehdr elf32_getphdr elf32_getshdr elf32_newehdr elf32_newphdr elf32_xlatetof elf32_xlatetom elf64_checksum elf64_fsize elf64_getehdr elf64_getphdr elf64_getshdr elf64_newehdr elf64_newphdr elf64_xlatetof elf64_xlatetom elf_begin elf_cntl elf_end elf_errmsg elf_errno elf_fill elf_flagdata elf_flagehdr elf_flagelf elf_flagphdr elf_flagscn elf_flagshdr elf_getarhdr elf_getarsym elf_getbase elf_getdata elf_getident elf_getscn elf_hash elf_kind elf_memory elf_ndxscn elf_newdata elf_newscn elf_next elf_nextscn elf_rand elf_rawdata</pre>

elf_rawfile	elf_strptr
elf_update	elf_version
gelf_checksum	gelf_fsize
gelf_getclass	gelf_getdyn
gelf_getehdr	gelf_getmove
gelf_getphdr	gelf_getrel
gelf_getrela	gelf_getshdr
gelf_getsym	gelf_getsyminfo
gelf_newehdr	gelf_newphdr
gelf_update_dyn	gelf_update_ehdr
gelf_update_move	gelf_update_phdr
gelf_update_rel	gelf_update_rela
gelf_update_shdr	gelf_update_sym
gelf_update_syminfo	gelf_xlatetof
gelf_xlatetom	nlist

FILES /usr/lib/libelf.a archive library
 /usr/lib/libelf.so.1 shared object
 /usr/lib/64/libelf.so.1 64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl, SUNWarc (32-bit) SUNWcslx (64-bit)
MT-Level	Safe

SEE ALSO pvs(1), intro(3), elf(3ELF), gelf(3ELF), attributes(5)

libexacct(3LIB)

NAME	libexacct – extended accounting file access library
SYNOPSIS	<pre>cc [<i>flag...</i>] <i>file...</i> -lexacct [<i>library...</i>] #include <exacct.h></pre>
DESCRIPTION	Functions in this library define the interface for reading and writing extended accounting (<i>exacct</i>) files. The <i><exacct.h></i> header provides type and function declarations for all library services, as well as for the characteristics of accounting files generated by the Solaris kernel.
INTERFACES	The shared object <i>libexacct.so.1</i> provides the public interfaces defined below. See <i>intro(3)</i> for additional information on shared object interfaces.
	<pre>ea_alloc ea_attach_to_group ea_attach_to_object ea_close ea_copy_object ea_copy_object_tree ea_error ea_free ea_free_item ea_free_object ea_get_creator ea_get_hostname ea_get_object ea_get_object_tree ea_match_object_catalog ea_next_object ea_open ea_pack_object ea_previous_object ea_set_group ea_set_item ea_strdup ea_strfree ea_unpack_object ea_write_object</pre>
FILES	<pre>/usr/lib/libexacct.so.1 shared object /usr/lib/64/libexacct.so.1 64-bit shared object</pre>
ATTRIBUTES	See <i>attributes(5)</i> for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Evolving
MT-Level	MT-Safe

SEE ALSO acctadm(1M), intro(3), ea_error(3EXACCT), ea_open(3EXACCT), ea_pack_object(3EXACCT), ea_set_item(3EXACCT), attributes(5)

NOTES The SUNWosdem package provides source code for the exdump utility that uses the libexacct APIs to dump the contents of extended accounting files. The source code can be compiled in the directory /usr/demo/libexacct.

libform(3LIB)

NAME	libform – forms library																																																
SYNOPSIS	cc [<i>flag</i> . . .] <i>file</i> . . . -lform [<i>library</i> . . .]																																																
DESCRIPTION	Functions in this library provide forms using <code>libcurses(3LIB)</code> routines.																																																
INTERFACES	The shared object <code>libform.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.																																																
	<table><tr><td><code>current_field</code></td><td><code>data_ahead</code></td></tr><tr><td><code>data_behind</code></td><td><code>dup_field</code></td></tr><tr><td><code>dynamic_field_info</code></td><td><code>field_arg</code></td></tr><tr><td><code>field_back</code></td><td><code>field_buffer</code></td></tr><tr><td><code>field_count</code></td><td><code>field_fore</code></td></tr><tr><td><code>field_index</code></td><td><code>field_info</code></td></tr><tr><td><code>field_init</code></td><td><code>field_just</code></td></tr><tr><td><code>field_opts</code></td><td><code>field_opts_off</code></td></tr><tr><td><code>field_opts_on</code></td><td><code>field_pad</code></td></tr><tr><td><code>field_status</code></td><td><code>field_term</code></td></tr><tr><td><code>field_type</code></td><td><code>field_userptr</code></td></tr><tr><td><code>form_driver</code></td><td><code>form_fields</code></td></tr><tr><td><code>form_init</code></td><td><code>form_opts</code></td></tr><tr><td><code>form_opts_off</code></td><td><code>form_opts_on</code></td></tr><tr><td><code>form_page</code></td><td><code>form_sub</code></td></tr><tr><td><code>form_term</code></td><td><code>form_userptr</code></td></tr><tr><td><code>form_win</code></td><td><code>free_field</code></td></tr><tr><td><code>free_fieldtype</code></td><td><code>free_form</code></td></tr><tr><td><code>link_field</code></td><td><code>link_fieldtype</code></td></tr><tr><td><code>move_field</code></td><td><code>new_field</code></td></tr><tr><td><code>new_fieldtype</code></td><td><code>new_form</code></td></tr><tr><td><code>new_page</code></td><td><code>pos_form_cursor</code></td></tr><tr><td><code>post_form</code></td><td><code>scale_form</code></td></tr><tr><td><code>set_current_field</code></td><td><code>set_field_back</code></td></tr></table>	<code>current_field</code>	<code>data_ahead</code>	<code>data_behind</code>	<code>dup_field</code>	<code>dynamic_field_info</code>	<code>field_arg</code>	<code>field_back</code>	<code>field_buffer</code>	<code>field_count</code>	<code>field_fore</code>	<code>field_index</code>	<code>field_info</code>	<code>field_init</code>	<code>field_just</code>	<code>field_opts</code>	<code>field_opts_off</code>	<code>field_opts_on</code>	<code>field_pad</code>	<code>field_status</code>	<code>field_term</code>	<code>field_type</code>	<code>field_userptr</code>	<code>form_driver</code>	<code>form_fields</code>	<code>form_init</code>	<code>form_opts</code>	<code>form_opts_off</code>	<code>form_opts_on</code>	<code>form_page</code>	<code>form_sub</code>	<code>form_term</code>	<code>form_userptr</code>	<code>form_win</code>	<code>free_field</code>	<code>free_fieldtype</code>	<code>free_form</code>	<code>link_field</code>	<code>link_fieldtype</code>	<code>move_field</code>	<code>new_field</code>	<code>new_fieldtype</code>	<code>new_form</code>	<code>new_page</code>	<code>pos_form_cursor</code>	<code>post_form</code>	<code>scale_form</code>	<code>set_current_field</code>	<code>set_field_back</code>
<code>current_field</code>	<code>data_ahead</code>																																																
<code>data_behind</code>	<code>dup_field</code>																																																
<code>dynamic_field_info</code>	<code>field_arg</code>																																																
<code>field_back</code>	<code>field_buffer</code>																																																
<code>field_count</code>	<code>field_fore</code>																																																
<code>field_index</code>	<code>field_info</code>																																																
<code>field_init</code>	<code>field_just</code>																																																
<code>field_opts</code>	<code>field_opts_off</code>																																																
<code>field_opts_on</code>	<code>field_pad</code>																																																
<code>field_status</code>	<code>field_term</code>																																																
<code>field_type</code>	<code>field_userptr</code>																																																
<code>form_driver</code>	<code>form_fields</code>																																																
<code>form_init</code>	<code>form_opts</code>																																																
<code>form_opts_off</code>	<code>form_opts_on</code>																																																
<code>form_page</code>	<code>form_sub</code>																																																
<code>form_term</code>	<code>form_userptr</code>																																																
<code>form_win</code>	<code>free_field</code>																																																
<code>free_fieldtype</code>	<code>free_form</code>																																																
<code>link_field</code>	<code>link_fieldtype</code>																																																
<code>move_field</code>	<code>new_field</code>																																																
<code>new_fieldtype</code>	<code>new_form</code>																																																
<code>new_page</code>	<code>pos_form_cursor</code>																																																
<code>post_form</code>	<code>scale_form</code>																																																
<code>set_current_field</code>	<code>set_field_back</code>																																																

set_field_buffer	set_field_fore
set_field_init	set_field_just
set_field_opts	set_field_pad
set_field_status	set_field_term
set_field_type	set_field_userptr
set_fieldtype_arg	set_fieldtype_choice
set_form_fields	set_form_init
set_form_opts	set_form_page
set_form_sub	set_form_term
set_form_userptr	set_form_win
set_max_field	set_new_page
unpost_form	

FILES

/usr/lib/libform.a	archive library
/usr/lib/libform.so.1	shared object
/usr/lib/64/libform.so.1	64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl, SUNWarc (32-bit) SUNWcslx (64-bit)
MT-Level	Unsafe

SEE ALSO intro(3), libcurses(3LIB), attributes(5)

libgen(3LIB)

NAME libgen – string pattern-matching library

SYNOPSIS `cc [flag . . .] file . . . -lgen [library . . .]`

DESCRIPTION Functions in this library provide routines for string pattern-matching and pathname manipulation.

INTERFACES The shared object `libgen.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

<code>__braelist</code>	<code>__braslist</code>	<code>__loc1</code>
<code>__loc2</code>	<code>__locs</code>	<code>__nbra</code>
<code>__regerrno</code>	<code>__reglength</code>	<code>advance</code>
<code>bgets</code>	<code>braelist</code>	<code>braslist</code>
<code>bufsplit</code>	<code>compile</code>	<code>copylist</code>
<code>eaccess</code>	<code>gmatch</code>	<code>isencrypt</code>
<code>loc1</code>	<code>loc2</code>	<code>locs</code>
<code>mkdirp</code>	<code>nbra</code>	<code>p2close</code>
<code>p2open</code>	<code>pathfind</code>	<code>regerrno</code>
<code>reglength</code>	<code>rmdirp</code>	<code>step</code>
<code>strcadd</code>	<code>strecpy</code>	<code>streadd</code>
<code>strecpy</code>	<code>strfind</code>	<code>strrspn</code>
<code>strtrns</code>		

The following interface is unique to the 32-bit version of this library:

`copylist64`

FILES

<code>/usr/lib/libgen.a</code>	archive library
<code>/usr/lib/libgen.so.1</code>	shared object
<code>/usr/lib/64/libgen.so.1</code>	64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl, SUNWarc (32-bit)

libgen(3LIB)

ATTRIBUTE TYPE	ATTRIBUTE VALUE
	SUNWcslx (64-bit)
MT-Level	Safe

SEE ALSO `intro(3)`, `attributes(5)`

libgss(3LIB)

NAME	libgss – Generic Security Services library
SYNOPSIS	<pre>cc [<i>flag</i> . . .] <i>file</i> . . . -lgss [<i>library</i> . . .] #include <gssapi/gssapi.h></pre>
DESCRIPTION	The functions in this library are the routines that comprise the Generic Security Services library.
INTERFACES	The shared object <code>libgss.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.
	<pre>GSS_C_NT_ANONYMOUS GSS_C_NT_EXPORT_NAME GSS_C_NT_HOSTBASED_SERVICE GSS_C_NT_MACHINE_UID_NAME GSS_C_NT_STRING_UID_NAME GSS_C_NT_USER_NAME gss_accept_sec_context gss_acquire_cred gss_add_cred gss_add_oid_set_member gss_canonicalize_name gss_compare_name gss_context_time gss_create_empty_oid_set gss_delete_sec_context gss_display_name gss_display_status gss_duplicate_name gss_export_name gss_export_sec_context gss_get_mic gss_import_name gss_import_sec_context gss_indicate_mechs gss_init_sec_context gss_inquire_context gss_inquire_cred gss_inquire_cred_by_mech gss_inquire_mechs_for_name gss_inquire_names_for_mech gss_process_context_token gss_release_buffer gss_release_cred gss_release_name gss_release_oid gss_release_oid_set gss_seal gss_sign gss_test_oid_set_member gss_unseal gss_unwrap gss_verify gss_verify_mic gss_wrap gss_wrap_size_limit</pre>

FILES /usr/lib/libgss.so.1
shared object

/usr/lib/64/libgss.so.1
64-bit shared object file

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWgss (32-bit) SUNWgssx (64-bit)
MT-Level	Safe

SEE ALSO `pvs(1)`, `intro(2)`, `intro(3)`, `attributes(5)`
GSS-API Programming Guide

libidnkit(3LIB)

NAME	libidnkit – IDN conversion library								
SYNOPSIS	cc [<i>flag...</i>] <i>file...</i> -lidnkit [<i>library...</i>] #include <idn/api.h>								
DESCRIPTION	<p>Functions in this library provide conversions between ACE string and multibyte character string of the current locale or a specified codeset. They support various manipulations of internationalized domain names, including encoding conversion and name preparation. They are designed according to IDNA framework where each application must do necessary preparations for the internationalized domain names before passing them to the resolver. The library provides easy-to-use, high-level interfaces to help applications with the preparation.</p> <p>The libidnkit library internally uses iconv(3C) to provide encoding conversion from UTF-8 to the local encoding (such as ISO8859-1, usually determined by the current locale), and from the local encoding to UTF-8.</p>								
INTERFACES	<p>The shared object libidnkit.so.1 provides the public interfaces defined below. See intro(3) for additional information on shared object interfaces.</p> <pre> idn_decodename idn_decodename2 idn_enable idn_encodename idn_nameinit </pre>								
FILES	<pre> /usr/lib/libidnkit.so.1 shared object /usr/lib/64/libidnkit.so.1 64-bit shared object </pre>								
ATTRIBUTES	See attributes(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>Availability</td> <td>SUNWidnl</td> </tr> <tr> <td>Interface Stability</td> <td>Evolving</td> </tr> <tr> <td>MT-Level</td> <td>Unsafe</td> </tr> </tbody> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	Availability	SUNWidnl	Interface Stability	Evolving	MT-Level	Unsafe
ATTRIBUTE TYPE	ATTRIBUTE VALUE								
Availability	SUNWidnl								
Interface Stability	Evolving								
MT-Level	Unsafe								
SEE ALSO	<p>intro(3), iconv(3C), idn_decodename(3EXT), setlocale(3C), hosts(4), attributes(5), environ(5)</p> <p>RFC 3490 Internationalizing Domain Names in Applications (IDNA)</p> <p>RFC 3491 Nameprep: A Stringprep Profile for Internationalized Domain Names (IDN)</p> <p>RFC 3492 Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)</p>								

RFC 3454 Preparation of Internationalized Strings ("stringprep")
 RFC 952 DoD Internet Host Table Specification
 RFC 921 Domain Name System Implementation Schedule - Revised
 STD 3, RFC 1122 Requirements for Internet Hosts -- Communication Layers
 STD 3, RFC 1123 Requirements for Internet Hosts -- Applications and Support

Unicode Standard Annex #15: Unicode Normalization Forms, Version 3.2.0.
<http://www.unicode.org/unicode/reports/tr15/tr15-22.html>

International Language Environments Guide (for this version of Solaris)

**COPYRIGHT
AND LICENSE**

Copyright (c) 2000-2002 Japan Network Information Center. All rights reserved.

By using this file, you agree to the terms and conditions set forth below.

LICENSE TERMS AND CONDITIONS

The following License Terms and Conditions apply, unless a different license is obtained from Japan Network Information Center ("JPNIC"), a Japanese association, Kokusai-Kougyou-Kanda Bldg 6F, 2-3-4 Uchi-Kanda, Chiyoda-ku, Tokyo 101-0047, Japan.

1. Use, Modification and Redistribution (including distribution of any modified or derived work) in source and/or binary forms is permitted under this License Terms and Conditions.
2. Redistribution of source code must retain the copyright notices as they appear in each source code file, this License Terms and Conditions.
3. Redistribution in binary form must reproduce the Copyright Notice, this License Terms and Conditions, in the documentation and/or other materials provided with the distribution. For the purposes of binary distribution the "Copyright Notice" refers to the following language: "Copyright (c) 2000-2002 Japan Network Information Center. All rights reserved."
4. The name of JPNIC may not be used to endorse or promote products derived from this Software without specific prior written approval of JPNIC.
5. Disclaimer/Limitation of Liability: THIS SOFTWARE IS PROVIDED BY JPNIC "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JPNIC BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF

libidnkit(3LIB)

ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

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(3LIB).

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.

INTERFACES The shared object libintl.so.1 provides the public interfaces defined below. See intro(3) for additional information on shared object interfaces.

bindtextdomain	dcgettext
dgettext	gettext
textdomain	

FILES

/usr/lib/libintl.a	a link to /usr/lib/null.a
/usr/lib/libintl.so.1	a filter on libc.so.1
/usr/lib/64/libintl.so.1	a filter on 64/libc.so.1

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl, SUNWarc (32-bit) SUNWcslx (64-bit)
MT-Level	Safe with exceptions

SEE ALSO pvs(1), intro(3), gettext(3C), libc(3LIB), attributes(5)

libkstat(3LIB)

NAME	libkstat – kernel statistics library
SYNOPSIS	<pre>cc [<i>flag</i> . . .] <i>file</i> . . . -lkstat [<i>library</i> . . .] #include <kstat.h></pre>
DESCRIPTION	Functions in this library provide a general-purpose mechanism for providing kernel statistics to users.
INTERFACES	The shared object <code>libkstat.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.
	<pre>kstat_chain_update kstat_close kstat_data_lookup kstat_lookup kstat_open kstat_read kstat_write</pre>
FILES	<pre>/usr/lib/libkstat.so.1 shared object /usr/lib/64/libkstat.so.1 64-bit shared object</pre>
ATTRIBUTES	See <code>attributes(5)</code> for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)
MT-Level	Unsafe

SEE ALSO `pvs(1)`, `intro(3)`, `kstat(3KSTAT)`, `attributes(5)`

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 functions are documented in Section 3KVM of the reference manual.

All `libkvm` functions are Uncommitted, since there is almost nothing that can be put as a symbol in a namelist that has stability from release to release. The syntax of these functions has been stable from release to release, but this is subject to change.

INTERFACES The shared object `libkvm.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

<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
`/usr/lib/64/libkvm.so.1` 64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl (32-bit) SUNWcsl x(64-bit)
MT-Level	Unsafe

SEE ALSO `pvs(1)`, `intro(3)`, `attributes(5)`

libl(3LIB)

NAME	libl – lex library						
SYNOPSIS	cc [<i>flag</i> . . .] <i>file</i> . . . -ll [<i>library</i> . . .]						
DESCRIPTION	Functions in this library provide user interfaces to the lex(1) library.						
INTERFACES	The shared object libl.so.1 provides the public interfaces defined below. See intro(3) for additional information on shared object interfaces. allprint allprint_w main sprint sprint_w yyless yyless_e yyless_w yyracc yyreject yyreject_e yyreject_w yywrap						
FILES	/usr/lib/libl.a archive library /usr/lib/libl.so.1 shared object /usr/lib/64/libl.so.1 64-bit shared object						
ATTRIBUTES	See attributes(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>Availability</td><td>SUNWcsl (32-bit) SUNWcslx (64-bit)</td></tr><tr><td>MT-Level</td><td>Unsafe</td></tr></tbody></table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)	MT-Level	Unsafe
ATTRIBUTE TYPE	ATTRIBUTE VALUE						
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)						
MT-Level	Unsafe						
SEE ALSO	lex(1), intro(3), attributes(5)						

NAME liblayout – layout service library

SYNOPSIS `cc [flag . . .] file . . . -llayout [library . . .]`
`#include <sys/layout.h>`

DESCRIPTION Functions in this library provide various layout service routines.

INTERFACES The shared object `liblayout.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

`m_create_layout` `m_destroy_layout`
`m_getvalues_layout` `m_setvalues_layout`
`m_transform_layout` `m_wtransform_layout`

FILES `/usr/lib/liblayout.so.1` shared object
`/usr/lib/64/liblayout.so.1` 64-bit shared object.

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWctpls (32-bit) SUNWctplx (64-bit)
MT Level	MT-Safe

SEE ALSO `intro(3)`, `attributes(5)`

liblgrp(3LIB)

NAME	liblgrp – locality group library																
SYNOPSIS	cc [<i>flag...</i>] <i>file...</i> -llgrp [<i>library...</i>] #include <sys/lgrp_user.h>																
DESCRIPTION	The functions in this library traverse the lgroup (locality group) hierarchy, discover its contents, and set a thread's affinity for an lgroup. A locality group represents the set of CPU-like and memory-like hardware devices that are at most some locality apart from each other.																
INTERFACES	The shared object liblgrp.so.1 provides the public interfaces defined below. See intro(3) for additional information on shared object interfaces.																
	<table border="0"> <tr> <td>lgrp_affinity_get</td> <td>lgrp_affinity_set</td> </tr> <tr> <td>lgrp_children</td> <td>lgrp_cookie_stale</td> </tr> <tr> <td>lgrp_cpus</td> <td>lgrp_fini</td> </tr> <tr> <td>lgrp_home</td> <td>lgrp_init</td> </tr> <tr> <td>lgrp_latency</td> <td>lgrp_mem_size</td> </tr> <tr> <td>lgrp_nlgrps</td> <td>lgrp_parents</td> </tr> <tr> <td>lgrp_root</td> <td>lgrp_version</td> </tr> <tr> <td>lgrp_view</td> <td></td> </tr> </table>	lgrp_affinity_get	lgrp_affinity_set	lgrp_children	lgrp_cookie_stale	lgrp_cpus	lgrp_fini	lgrp_home	lgrp_init	lgrp_latency	lgrp_mem_size	lgrp_nlgrps	lgrp_parents	lgrp_root	lgrp_version	lgrp_view	
lgrp_affinity_get	lgrp_affinity_set																
lgrp_children	lgrp_cookie_stale																
lgrp_cpus	lgrp_fini																
lgrp_home	lgrp_init																
lgrp_latency	lgrp_mem_size																
lgrp_nlgrps	lgrp_parents																
lgrp_root	lgrp_version																
lgrp_view																	
FILES	<table border="0"> <tr> <td>/usr/lib/liblgrp.so.1</td> <td>shared object</td> </tr> <tr> <td>/usr/lib/64/liblgrp.so.1</td> <td>64-bit shared object</td> </tr> </table>	/usr/lib/liblgrp.so.1	shared object	/usr/lib/64/liblgrp.so.1	64-bit shared object												
/usr/lib/liblgrp.so.1	shared object																
/usr/lib/64/liblgrp.so.1	64-bit shared object																
ATTRIBUTES	See attributes(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>Availability</td> <td>SUNWcsl (32-bit) SUNWcslx (64-bit)</td> </tr> <tr> <td>Interface Stability</td> <td>Evolving</td> </tr> <tr> <td>MT-Level</td> <td>MT-Safe</td> </tr> </tbody> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)	Interface Stability	Evolving	MT-Level	MT-Safe								
ATTRIBUTE TYPE	ATTRIBUTE VALUE																
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)																
Interface Stability	Evolving																
MT-Level	MT-Safe																
SEE ALSO	intro(3), lgrp_affinity_get(3LGRP), lgrp_children(3LGRP), lgrp_cookie_stale(3LGRP), lgrp_cpus(3LGRP), lgrp_fini(3LGRP), lgrp_home(3LGRP), lgrp_init(3LGRP), lgrp_latency(3LGRP), lgrp_mem_size(3LGRP), lgrp_nlgrps(3LGRP), lgrp_parents(3LGRP), lgrp_root(3LGRP), lgrp_version(3LGRP), lgrp_view(3LGRP), attributes(5)																

NAME libmail – user mailbox lockfile management library

SYNOPSIS `cc [flag . . .] file . . . -lmail [library . . .]`
`#include <maillock.h>`

DESCRIPTION Interfaces in this library provide functions for managing user mailbox lockfiles.

INTERFACES The shared object `libmail.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

`maillock` `mailunlock` `touchlock`

FILES

`/usr/lib/libmail.a`
archive library

`/usr/lib/libmail.so.1`
shared object

`/usr/lib/64/libmail.so.1`
64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)
MT Level	Unsafe

SEE ALSO `intro(3)`, `maillock(3MAIL)`, `attributes(5)`

libmalloc(3LIB)

NAME	libmalloc – memory allocation library												
SYNOPSIS	cc [<i>flag</i> . . .] <i>file</i> . . . -lmalloc [<i>library</i> . . .]												
DESCRIPTION	Functions in this library provide routines for memory allocation. These routines are space-efficient but have lower performance. Their usage can result in serious performance degradation.												
INTERFACES	The shared object <code>libmalloc.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces. <table><tr><td><code>_cfree</code></td><td><code>_mallinfo</code></td><td><code>_mallopt</code></td><td><code>calloc</code></td></tr><tr><td><code>cfree</code></td><td><code>free</code></td><td><code>mallinfo</code></td><td><code>malloc</code></td></tr><tr><td><code>mallopt</code></td><td><code>realloc</code></td><td></td><td></td></tr></table>	<code>_cfree</code>	<code>_mallinfo</code>	<code>_mallopt</code>	<code>calloc</code>	<code>cfree</code>	<code>free</code>	<code>mallinfo</code>	<code>malloc</code>	<code>mallopt</code>	<code>realloc</code>		
<code>_cfree</code>	<code>_mallinfo</code>	<code>_mallopt</code>	<code>calloc</code>										
<code>cfree</code>	<code>free</code>	<code>mallinfo</code>	<code>malloc</code>										
<code>mallopt</code>	<code>realloc</code>												
FILES	<code>/usr/lib/libmalloc.a</code> archive library <code>/usr/lib/libmalloc.so.1</code> shared object <code>/usr/lib/64/libmalloc.so.1</code> 64-bit shared object												
ATTRIBUTES	See <code>attributes(5)</code> for descriptions of the following attributes: <table border="1"><thead><tr><th>ATTRIBUTE TYPE</th><th>ATTRIBUTE VALUE</th></tr></thead><tbody><tr><td>Availability</td><td>SUNWcsl (32-bit) SUNWcslx (64-bit)</td></tr><tr><td>MT-Level</td><td>Safe</td></tr></tbody></table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)	MT-Level	Safe						
ATTRIBUTE TYPE	ATTRIBUTE VALUE												
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)												
MT-Level	Safe												
SEE ALSO	<code>intro(3)</code> , <code>malloc(3MALLOC)</code> , <code>attributes(5)</code>												

NAME libmapmalloc – alternative memory allocator library

SYNOPSIS `cc [flag . . .] file . . . -lmapmalloc [library . . .]`
`#include <stdlib.h>`

DESCRIPTION Functions in this library provide `malloc` routines that use `mmap(2)` instead of `sbrk(2)` for acquiring heap space.

INTERFACES The shared object `libmapmalloc.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

`calloc` `cfree` `free` `mallinfo` `malloc`
`mallopt` `memalign` `realloc` `valloc`

FILES `/usr/lib/libmapmalloc.a`
archive library
`/usr/lib/libmapmalloc.so.1`
shared object
`/usr/lib/64/libmapmalloc.so.1`
64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)
MT-Level	Safe

SEE ALSO `pvs(1)`, `mmap(2)`, `sbrk(2)`, `intro(3)`, `malloc(3C)`, `malloc(3MALLOC)`, `mapmalloc(3MALLOC)`, `attributes(5)`

libmd5(3LIB)

NAME	libmd5 – MD5 hashing library
SYNOPSIS	<pre>cc [<i>flag</i> . . .] <i>file</i> . . . -lmd5 [<i>library</i> . . .] #include <md5.h></pre>
DESCRIPTION	Functions in this library provide MD5 hashing routines.
INTERFACES	The shared object <code>libmd5.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.
	<pre>MD5Final MD5Init MD5Update md5_calc</pre>
FILES	<pre>/usr/lib/libmd5.so.1 shared object /usr/lib/64/libmd5.so.1 64-bit shared object.</pre>
ATTRIBUTES	See <code>attributes(5)</code> for description of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)
MT Level	MT-Safe

SEE ALSO `intro(3)`, `attributes(5)`

NAME	libmenu – menus library	
SYNOPSIS	cc [<i>flag</i> . . .] <i>file</i> . . . -lmenu [<i>library</i> . . .]	
DESCRIPTION	Functions in this library provide menus using libcurses(3LIB) routines.	
INTERFACES	The shared object libmenu.so.1 provides the public interfaces defined below. See intro(3) for additional information on shared object interfaces.	
	current_item	free_item
	free_menu	item_count
	item_description	item_index
	item_init	item_name
	item_opts	item_opts_off
	item_opts_on	item_term
	item_userptr	item_value
	item_visible	menu_back
	menu_driver	menu_fore
	menu_format	menu_grey
	menu_init	menu_items
	menu_mark	menu_opts
	menu_opts_off	menu_opts_on
	menu_pad	menu_pattern
	menu_sub	menu_term
	menu_userptr	menu_win
	new_item	new_menu
	pos_menu_cursor	post_menu
	scale_menu	set_current_item
	set_item_init	set_item_opts
	set_item_term	set_item_userptr
	set_item_value	set_menu_back
	set_menu_fore	set_menu_format
	set_menu_grey	set_menu_init

libmenu(3LIB)

set_menu_items	set_menu_mark
set_menu_opts	set_menu_pad
set_menu_pattern	set_menu_sub
set_menu_term	set_menu_userptr
set_menu_win	set_top_row
top_row	unpost_menu

FILES /usr/lib/libmenu.a archive library
/usr/lib/libmenu.so.1 shared object
/usr/lib/64/libmenu.so.1 64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)
MT-Level	Unsafe

SEE ALSO [intro\(3\)](#), [libcurses\(3LIB\)](#), [attributes\(5\)](#)

NAME	libmp – multiple precision library							
SYNOPSIS	<pre>cc [<i>flag</i> . . .] <i>file</i> . . . -lmp [<i>library</i> . . .] #include <mp.h></pre>							
DESCRIPTION	Functions in this library provide various multiple precision routines.							
INTERFACES	The shared object <code>libmp.so.2</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.							
	<pre>mp_gcd mp_itom mp_madd mp_mcmp mp_mdiv mp_mfree mp_min mp_mout mp_msqrt mp_msub mp_mtox mp_mult mp_pow mp_rpow mp_sdiv mp_xtom</pre>							
FILES	<pre>/usr/lib/libmp.a /usr/lib/libmp.so.1 /usr/lib/libmp.so.2 /usr/lib/64/libmp.so.2</pre>	<pre>archive library shared object for binary compatibility only shared object 64-bit shared object</pre>						
ATTRIBUTES	See <code>attributes(5)</code> for descriptions of the following attributes:							
	<table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>Availability</td> <td>SUNWcsl (32-bit) SUNWcslx (64-bit)</td> </tr> <tr> <td>MT-Level</td> <td>Unsafe</td> </tr> </tbody> </table>		ATTRIBUTE TYPE	ATTRIBUTE VALUE	Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)	MT-Level	Unsafe
ATTRIBUTE TYPE	ATTRIBUTE VALUE							
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)							
MT-Level	Unsafe							
SEE ALSO	<code>pvs(1)</code> , <code>intro(3)</code> , <code>exp(3M)</code> , <code>mp(3MP)</code> , <code>attributes(5)</code>							

libmtmalloc(3LIB)

NAME	libmtmalloc – multi-threaded memory allocator library								
SYNOPSIS	<pre>cc [<i>flag...</i>] <i>file...</i> -lmtmalloc [<i>library...</i>] #include <mtmalloc.h></pre>								
DESCRIPTION	Functions in this library provide concurrent access to heap space.								
INTERFACES	The shared object <code>libmtmalloc.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces. <table><tr><td><code>calloc</code></td><td><code>free</code></td></tr><tr><td><code>malloc</code></td><td><code>mallocctl</code></td></tr><tr><td><code>memalign</code></td><td><code>realloc</code></td></tr><tr><td><code>valloc</code></td><td></td></tr></table>	<code>calloc</code>	<code>free</code>	<code>malloc</code>	<code>mallocctl</code>	<code>memalign</code>	<code>realloc</code>	<code>valloc</code>	
<code>calloc</code>	<code>free</code>								
<code>malloc</code>	<code>mallocctl</code>								
<code>memalign</code>	<code>realloc</code>								
<code>valloc</code>									
FILES	<pre>/usr/lib/libmtmalloc.so.1 shared object /usr/lib/64/libmtmalloc.so.1 64-bit shared object</pre>								
ATTRIBUTES	See <code>attributes(5)</code> for descriptions of the following attributes: <table border="1"><thead><tr><th>ATTRIBUTE TYPE</th><th>ATTRIBUTE VALUE</th></tr></thead><tbody><tr><td>Availability</td><td>SUNWcsl (32-bit) SUNWcslx (64-bit)</td></tr><tr><td>MT-Level</td><td>Safe</td></tr></tbody></table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)	MT-Level	Safe		
ATTRIBUTE TYPE	ATTRIBUTE VALUE								
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)								
MT-Level	Safe								
SEE ALSO	<code>pvs(1)</code> , <code>sbrk(2)</code> , <code>intro(3)</code> , <code>malloc(3C)</code> , <code>malloc(3MALLOC)</code> , <code>mapmalloc(3MALLOC)</code> , <code>mtmalloc(3MALLOC)</code> , <code>attributes(5)</code>								

NAME	libnsl – network services library																																
SYNOPSIS	<code>cc [<i>flag</i> . . .] <i>file</i> . . . -lnsl [<i>library</i> . . .]</code>																																
DESCRIPTION	<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>Many features in this library are implemented upon dynamic linking and will not function correctly if the library is statically linked. Additionally, an application that statically links this library will not be compliant with the System V Application Binary Interface.</p> <p>Some symbols are not intended to be referenced directly. Rather, they are exposed because they are used elsewhere through a private interface. One such example is the set of symbols beginning with the <code>_xti</code> prefix. Those symbols are used in implementing the X/Open Transport Interface (XTI) interfaces documented in <code>libxnet</code>. See <code>libxnet(3LIB)</code>.</p>																																
INTERFACES	<p>The shared object <code>libnsl.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.</p> <table border="0" style="width: 100%;"> <tr> <td><code>__rpc_createerr</code></td> <td><code>__t_errno</code></td> </tr> <tr> <td><code>_nderror</code></td> <td><code>_null_auth</code></td> </tr> <tr> <td><code>_xti_accept</code></td> <td><code>_xti_alloc</code></td> </tr> <tr> <td><code>_xti_bind</code></td> <td><code>_xti_close</code></td> </tr> <tr> <td><code>_xti_connect</code></td> <td><code>_xti_error</code></td> </tr> <tr> <td><code>_xti_free</code></td> <td><code>_xti_getinfo</code></td> </tr> <tr> <td><code>_xti_getprotaddr</code></td> <td><code>_xti_getstate</code></td> </tr> <tr> <td><code>_xti_listen</code></td> <td><code>_xti_look</code></td> </tr> <tr> <td><code>_xti_open</code></td> <td><code>_xti_optmgmt</code></td> </tr> <tr> <td><code>_xti_rcv</code></td> <td><code>_xti_rcvconnect</code></td> </tr> <tr> <td><code>_xti_rcvdis</code></td> <td><code>_xti_rcvrel</code></td> </tr> <tr> <td><code>_xti_rcvreldata</code></td> <td><code>_xti_rcvudata</code></td> </tr> <tr> <td><code>_xti_rcvuderr</code></td> <td><code>_xti_rcvv</code></td> </tr> <tr> <td><code>_xti_rcvvudata</code></td> <td><code>_xti_snd</code></td> </tr> <tr> <td><code>_xti_snddis</code></td> <td><code>_xti_sndrel</code></td> </tr> <tr> <td><code>_xti_sndreldata</code></td> <td><code>_xti_sndudata</code></td> </tr> </table>	<code>__rpc_createerr</code>	<code>__t_errno</code>	<code>_nderror</code>	<code>_null_auth</code>	<code>_xti_accept</code>	<code>_xti_alloc</code>	<code>_xti_bind</code>	<code>_xti_close</code>	<code>_xti_connect</code>	<code>_xti_error</code>	<code>_xti_free</code>	<code>_xti_getinfo</code>	<code>_xti_getprotaddr</code>	<code>_xti_getstate</code>	<code>_xti_listen</code>	<code>_xti_look</code>	<code>_xti_open</code>	<code>_xti_optmgmt</code>	<code>_xti_rcv</code>	<code>_xti_rcvconnect</code>	<code>_xti_rcvdis</code>	<code>_xti_rcvrel</code>	<code>_xti_rcvreldata</code>	<code>_xti_rcvudata</code>	<code>_xti_rcvuderr</code>	<code>_xti_rcvv</code>	<code>_xti_rcvvudata</code>	<code>_xti_snd</code>	<code>_xti_snddis</code>	<code>_xti_sndrel</code>	<code>_xti_sndreldata</code>	<code>_xti_sndudata</code>
<code>__rpc_createerr</code>	<code>__t_errno</code>																																
<code>_nderror</code>	<code>_null_auth</code>																																
<code>_xti_accept</code>	<code>_xti_alloc</code>																																
<code>_xti_bind</code>	<code>_xti_close</code>																																
<code>_xti_connect</code>	<code>_xti_error</code>																																
<code>_xti_free</code>	<code>_xti_getinfo</code>																																
<code>_xti_getprotaddr</code>	<code>_xti_getstate</code>																																
<code>_xti_listen</code>	<code>_xti_look</code>																																
<code>_xti_open</code>	<code>_xti_optmgmt</code>																																
<code>_xti_rcv</code>	<code>_xti_rcvconnect</code>																																
<code>_xti_rcvdis</code>	<code>_xti_rcvrel</code>																																
<code>_xti_rcvreldata</code>	<code>_xti_rcvudata</code>																																
<code>_xti_rcvuderr</code>	<code>_xti_rcvv</code>																																
<code>_xti_rcvvudata</code>	<code>_xti_snd</code>																																
<code>_xti_snddis</code>	<code>_xti_sndrel</code>																																
<code>_xti_sndreldata</code>	<code>_xti_sndudata</code>																																

libnsl(3LIB)

<code>_xti_sndv</code>	<code>_xti_sndvudata</code>
<code>_xti_strerror</code>	<code>_xti_sync</code>
<code>_xti_sysconf</code>	<code>_xti_unbind</code>
<code>_xti_xns5_accept</code>	<code>_xti_xns5_snd</code>
<code>auth_destroy</code>	<code>authdes_create</code>
<code>authdes_getucred</code>	<code>authdes_lock</code>
<code>authdes_seccreate</code>	<code>authnone_create</code>
<code>authsys_create</code>	<code>authsys_create_default</code>
<code>callrpc</code>	<code>clnt_broadcast</code>
<code>clnt_call</code>	<code>clnt_control</code>
<code>clnt_create</code>	<code>clnt_create_timed</code>
<code>clnt_create_vers</code>	<code>clnt_create_vers_timed</code>
<code>clnt_destroy</code>	<code>clnt_dg_create</code>
<code>clnt_door_create</code>	<code>clnt_freeres</code>
<code>clnt_geterr</code>	<code>clnt_pcreateerror</code>
<code>clnt_perrno</code>	<code>clnt_perror</code>
<code>clnt_raw_create</code>	<code>clnt_spcreateerror</code>
<code>clnt_sperrno</code>	<code>clnt_sperror</code>
<code>clnt_tli_create</code>	<code>clnt_tp_create</code>
<code>clnt_tp_create_timed</code>	<code>clnt_vc_create</code>
<code>clntraw_create</code>	<code>clnttcp_create</code>
<code>clntudp_bufcreate</code>	<code>clntudp_create</code>
<code>dbmclose</code>	<code>dbminit</code>
<code>delete</code>	<code>des_setparity</code>
<code>dial</code>	<code>doconfig</code>
<code>endhostent</code>	<code>endnetconfig</code>
<code>endnetpath</code>	<code>endrpcent</code>
<code>fetch</code>	<code>firstkey</code>
<code>freehostent</code>	<code>freenetconfigt</code>
<code>get_myaddress</code>	<code>gethostbyaddr</code>

gethostbyaddr_r	gethostbyname
gethostbyname_r	gethostent
gethostent_r	getipnodebyaddr
getipnodebyname	getnetconfig
getnetconfigent	getnetname
getnetpath	getpublickey
getrpcbyname	getrpcbyname_r
getrpcbynumber	getrpcbynumber_r
getrpccent	getrpccent_r
getrpcport	getsecretkey
h_errno	host2netname
inet_addr	inet_netof
inet_ntoa	inet_ntoa_r
inet_ntop	inet_pton
key_decryptsession	key_encryptsession
key_gendes	key_secretkey_is_set
key_setsecret	maxbno
nc_perror	nc_sperror
netdir_free	netdir_getbyaddr
netdir_getbyname	netdir_options
netdir_perror	netdir_sperror
netname2host	netname2user
nextkey	nis_add
nis_add_entry	nis_addmember
nis_checkpoint	nis_clone_object
nis_creategroup	nis_data
nis_destroy_object	nis_destroygroup
nis_dir_cmp	nis_domain_of
nis_dump	nis_dumplog
nis_find_item	nis_finddirectory

libnsl(3LIB)

<code>nis_first_entry</code>	<code>nis_free_request</code>
<code>nis_freenames</code>	<code>nis_freeresult</code>
<code>nis_freeservlist</code>	<code>nis_freetags</code>
<code>nis_get_request</code>	<code>nis_get_static_storage</code>
<code>nis_getnames</code>	<code>nis_getservlist</code>
<code>nis_in_table</code>	<code>nis_insert_item</code>
<code>nis_insert_name</code>	<code>nis_ismember</code>
<code>nis_leaf_of</code>	<code>nis_leaf_of_r</code>
<code>nis_lerror</code>	<code>nis_list</code>
<code>nis_local_directory</code>	<code>nis_local_group</code>
<code>nis_local_host</code>	<code>nis_local_principal</code>
<code>nis_lookup</code>	<code>nis_make_error</code>
<code>nis_make_rpchandle</code>	<code>nis_mkdir</code>
<code>nis_modify</code>	<code>nis_modify_entry</code>
<code>nis_name_of</code>	<code>nis_next_entry</code>
<code>nis_perror</code>	<code>nis_ping</code>
<code>nis_print_directory</code>	<code>nis_print_entry</code>
<code>nis_print_group</code>	<code>nis_print_group_entry</code>
<code>nis_print_link</code>	<code>nis_print_object</code>
<code>nis_print_rights</code>	<code>nis_print_table</code>
<code>nis_read_obj</code>	<code>nis_remove</code>
<code>nis_remove_entry</code>	<code>nis_remove_item</code>
<code>nis_remove_name</code>	<code>nis_removemember</code>
<code>nis_rmdir</code>	<code>nis_servstate</code>
<code>nis_sperrno</code>	<code>nis_sperror</code>
<code>nis_sperror_r</code>	<code>nis_stats</code>
<code>nis_verifygroup</code>	<code>nis_write_obj</code>
<code>pmap_getmaps</code>	<code>pmap_getport</code>
<code>pmap_rmtcall</code>	<code>pmap_set</code>
<code>pmap_unset</code>	<code>registerrpc</code>

rpc_broadcast	rpc_broadcast_exp
rpc_call	rpc_control
rpc_createerr	rpc_gss_get_error
rpc_gss_get_mech_info	rpc_gss_get_mechanisms
rpc_gss_get_principal_name	rpc_gss_get_versions
rpc_gss_getcred	rpc_gss_is_installed
rpc_gss_max_data_length	rpc_gss_mech_to_oid
rpc_gss_qop_to_num	rpc_gss_seccreate
rpc_gss_set_callback	rpc_gss_set_defaults
rpc_gss_set_svc_name	rpc_gss_svc_max_data_length
rpc_reg	rpcb_getaddr
rpcb_getmaps	rpcb_gettime
rpcb_rmtcall	rpcb_set
rpcb_unset	sethostent
setnetconfig	setnetpath
setrpcent	store
svc_auth_reg	svc_control
svc_create	svc_destroy
svc_dg_create	svc_dg_enablecache
svc_done	svc_door_create
svc_exit	svc_fd_create
svc_fdset	svc_freeargs
svc_get_local_cred	svc_getargs
svc_getreq	svc_getreq_common
svc_getreq_poll	svc_getreqset
svc_getrpccaller	svc_max_pollfd
svc_pollfd	svc_raw_create
svc_reg	svc_register
svc_run	svc_sendreply
svc_tli_create	svc_tp_create

libnsl(3LIB)

svc_unreg	svc_unregister
svc_vc_create	svcerr_auth
svcerr_decode	svcerr_noproc
svcerr_noprog	svcerr_progvers
svcerr_systemerr	svcerr_weakauth
svcfid_create	svccraw_create
svctcp_create	svccudp_bufcreate
svccudp_create	t_accept
t_alloc	t_bind
t_close	t_connect
t_errno	t_error
t_free	t_getinfo
t_getname	t_getstate
t_listen	t_lock
t_nerr	t_open
t_optmgmt	t_rcv
t_rcvconnect	t_rcvdis
t_rcvrel	t_rcvudata
t_rcvuderr	t_snd
t_snddis	t_sndrel
t_sndudata	t_strerror
t_sync	t_unbind
taddr2uaddr	uaddr2taddr
undial	user2netname
xdr_accepted_reply	xdr_array
xdr_authsys_parms	xdr_bool
xdr_bytes	xdr_callhdr
xdr_callmsg	xdr_char
xdr_destroy	xdr_double
xdr_enum	xdr_float

xdr_free	xdr_getpos
xdr_hyper	xdr_inline
xdr_int	xdr_int16_t
xdr_int32_t	xdr_int64_t
xdr_int8_t	xdr_long
xdr_longlong_t	xdr_opaque
xdr_opaque_auth	xdr_pointer
xdr_quadruple	xdr_reference
xdr_rejected_reply	xdr_replymsg
xdr_setpos	xdr_short
xdr_sizeof	xdr_string
xdr_u_char	xdr_u_hyper
xdr_u_int	xdr_u_long
xdr_u_longlong_t	xdr_u_short
xdr_uint16_t	xdr_uint32_t
xdr_uint64_t	xdr_uint8_t
xdr_union	xdr_vector
xdr_void	xdr_wrapstring
xdrmem_create	xdrrec_create
xdrrec_endofrecord	xdrrec_eof
xdrrec_readbytes	xdrrec_skiprecord
xdrstdio_create	xprt_register
xprt_unregister	yp_all
yp_bind	yp_first
yp_get_default_domain	yp_master
yp_match	yp_next
yp_order	yp_unbind
yp_update	yperr_string
ypprot_err	

The following interface is unique to the 32-bit version of this library:

libnsl(3LIB)

`_new_svc_fdset`

FILES `/usr/lib/libnsl.a` archive library
`/usr/lib/libnsl.so.1` shared object
`/usr/lib/64/libnsl.so.1` 64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl, SUNWarc (32-bit) SUNWcslx (64-bit)
MT-Level	Safe with exceptions

SEE ALSO `pvs(1)`, `intro(2)`, `intro(3)`, `libxnet(3LIB)`, `attributes(5)`

NAME	libnvpair – name-value pair library																																																
SYNOPSIS	<pre>cc [<i>flag</i> . . .] <i>file</i> . . . -lnvpair [<i>library</i> . . .] #include <libnvpair.h></pre>																																																
DESCRIPTION	Functions in this library provide various name-value pair routines.																																																
INTERFACES	The shared object <code>libnvpair.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.																																																
	<table> <tr> <td><code>nvlist_add_boolean</code></td> <td><code>nvlist_add_byte</code></td> </tr> <tr> <td><code>nvlist_add_byte_array</code></td> <td><code>nvlist_add_int16</code></td> </tr> <tr> <td><code>nvlist_add_int16_array</code></td> <td><code>nvlist_add_int32</code></td> </tr> <tr> <td><code>nvlist_add_int32_array</code></td> <td><code>nvlist_add_int64</code></td> </tr> <tr> <td><code>nvlist_add_int64_array</code></td> <td><code>nvlist_add_string</code></td> </tr> <tr> <td><code>nvlist_add_string_array</code></td> <td><code>nvlist_add_uint16</code></td> </tr> <tr> <td><code>nvlist_add_uint16_array</code></td> <td><code>nvlist_add_uint32</code></td> </tr> <tr> <td><code>nvlist_add_uint32_array</code></td> <td><code>nvlist_add_uint64</code></td> </tr> <tr> <td><code>nvlist_add_uint64_array</code></td> <td><code>nvlist_alloc</code></td> </tr> <tr> <td><code>nvlist_dup</code></td> <td><code>nvlist_free</code></td> </tr> <tr> <td><code>nvlist_lookup_boolean</code></td> <td><code>nvlist_lookup_byte</code></td> </tr> <tr> <td><code>nvlist_lookup_byte_array</code></td> <td><code>nvlist_lookup_int16</code></td> </tr> <tr> <td><code>nvlist_lookup_int16_array</code></td> <td><code>nvlist_lookup_int32</code></td> </tr> <tr> <td><code>nvlist_lookup_int32_array</code></td> <td><code>nvlist_lookup_int64</code></td> </tr> <tr> <td><code>nvlist_lookup_int64_array</code></td> <td><code>nvlist_lookup_string</code></td> </tr> <tr> <td><code>nvlist_lookup_string_array</code></td> <td><code>nvlist_lookup_uint16</code></td> </tr> <tr> <td><code>nvlist_lookup_uint16_array</code></td> <td><code>nvlist_lookup_uint32</code></td> </tr> <tr> <td><code>nvlist_lookup_uint32_array</code></td> <td><code>nvlist_lookup_uint64</code></td> </tr> <tr> <td><code>nvlist_lookup_uint64_array</code></td> <td><code>nvlist_next_nvpair</code></td> </tr> <tr> <td><code>nvlist_pack</code></td> <td><code>nvlist_remove</code></td> </tr> <tr> <td><code>nvlist_remove_all</code></td> <td><code>nvlist_size</code></td> </tr> <tr> <td><code>nvlist_unpack</code></td> <td><code>nvpair_name</code></td> </tr> <tr> <td><code>nvpair_type</code></td> <td><code>nvpair_value_byte</code></td> </tr> <tr> <td><code>nvpair_value_byte_array</code></td> <td><code>nvpair_value_int16</code></td> </tr> </table>	<code>nvlist_add_boolean</code>	<code>nvlist_add_byte</code>	<code>nvlist_add_byte_array</code>	<code>nvlist_add_int16</code>	<code>nvlist_add_int16_array</code>	<code>nvlist_add_int32</code>	<code>nvlist_add_int32_array</code>	<code>nvlist_add_int64</code>	<code>nvlist_add_int64_array</code>	<code>nvlist_add_string</code>	<code>nvlist_add_string_array</code>	<code>nvlist_add_uint16</code>	<code>nvlist_add_uint16_array</code>	<code>nvlist_add_uint32</code>	<code>nvlist_add_uint32_array</code>	<code>nvlist_add_uint64</code>	<code>nvlist_add_uint64_array</code>	<code>nvlist_alloc</code>	<code>nvlist_dup</code>	<code>nvlist_free</code>	<code>nvlist_lookup_boolean</code>	<code>nvlist_lookup_byte</code>	<code>nvlist_lookup_byte_array</code>	<code>nvlist_lookup_int16</code>	<code>nvlist_lookup_int16_array</code>	<code>nvlist_lookup_int32</code>	<code>nvlist_lookup_int32_array</code>	<code>nvlist_lookup_int64</code>	<code>nvlist_lookup_int64_array</code>	<code>nvlist_lookup_string</code>	<code>nvlist_lookup_string_array</code>	<code>nvlist_lookup_uint16</code>	<code>nvlist_lookup_uint16_array</code>	<code>nvlist_lookup_uint32</code>	<code>nvlist_lookup_uint32_array</code>	<code>nvlist_lookup_uint64</code>	<code>nvlist_lookup_uint64_array</code>	<code>nvlist_next_nvpair</code>	<code>nvlist_pack</code>	<code>nvlist_remove</code>	<code>nvlist_remove_all</code>	<code>nvlist_size</code>	<code>nvlist_unpack</code>	<code>nvpair_name</code>	<code>nvpair_type</code>	<code>nvpair_value_byte</code>	<code>nvpair_value_byte_array</code>	<code>nvpair_value_int16</code>
<code>nvlist_add_boolean</code>	<code>nvlist_add_byte</code>																																																
<code>nvlist_add_byte_array</code>	<code>nvlist_add_int16</code>																																																
<code>nvlist_add_int16_array</code>	<code>nvlist_add_int32</code>																																																
<code>nvlist_add_int32_array</code>	<code>nvlist_add_int64</code>																																																
<code>nvlist_add_int64_array</code>	<code>nvlist_add_string</code>																																																
<code>nvlist_add_string_array</code>	<code>nvlist_add_uint16</code>																																																
<code>nvlist_add_uint16_array</code>	<code>nvlist_add_uint32</code>																																																
<code>nvlist_add_uint32_array</code>	<code>nvlist_add_uint64</code>																																																
<code>nvlist_add_uint64_array</code>	<code>nvlist_alloc</code>																																																
<code>nvlist_dup</code>	<code>nvlist_free</code>																																																
<code>nvlist_lookup_boolean</code>	<code>nvlist_lookup_byte</code>																																																
<code>nvlist_lookup_byte_array</code>	<code>nvlist_lookup_int16</code>																																																
<code>nvlist_lookup_int16_array</code>	<code>nvlist_lookup_int32</code>																																																
<code>nvlist_lookup_int32_array</code>	<code>nvlist_lookup_int64</code>																																																
<code>nvlist_lookup_int64_array</code>	<code>nvlist_lookup_string</code>																																																
<code>nvlist_lookup_string_array</code>	<code>nvlist_lookup_uint16</code>																																																
<code>nvlist_lookup_uint16_array</code>	<code>nvlist_lookup_uint32</code>																																																
<code>nvlist_lookup_uint32_array</code>	<code>nvlist_lookup_uint64</code>																																																
<code>nvlist_lookup_uint64_array</code>	<code>nvlist_next_nvpair</code>																																																
<code>nvlist_pack</code>	<code>nvlist_remove</code>																																																
<code>nvlist_remove_all</code>	<code>nvlist_size</code>																																																
<code>nvlist_unpack</code>	<code>nvpair_name</code>																																																
<code>nvpair_type</code>	<code>nvpair_value_byte</code>																																																
<code>nvpair_value_byte_array</code>	<code>nvpair_value_int16</code>																																																

libnvpair(3LIB)

<code>nvpair_value_int16_array</code>	<code>nvpair_value_int32</code>
<code>nvpair_value_int32_array</code>	<code>nvpair_value_int64</code>
<code>nvpair_value_int64_array</code>	<code>nvpair_value_string</code>
<code>nvpair_value_string_array</code>	<code>nvpair_value_uint16</code>
<code>nvpair_value_uint16_array</code>	<code>nvpair_value_uint32</code>
<code>nvpair_value_uint32_array</code>	<code>nvpair_value_uint64</code>
<code>nvpair_value_uint64_array</code>	

FILES `/usr/lib/libnvpair.so.1`
shared object
`/usr/lib/64/libnvpair.so.1`
64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)
MT-Level	MT-Safe

SEE ALSO `intro(3)`, `libnvpair(3NVP AIR)`, `attributes(5)`

NAME libpam – PAM (Pluggable Authentication Module) library

SYNOPSIS `cc [flag . . .] file . . . -lpam [library . . .]`
`#include <security/pam_appl.h>`

DESCRIPTION Functions in this library provide routines for the Pluggable Authentication Module (PAM).

INTERFACES The shared object `libpam.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

<code>pam_acct_mgmt</code>	<code>pam_authenticate</code>
<code>pam_chauthtok</code>	<code>pam_close_session</code>
<code>pam_end</code>	<code>pam_get_data</code>
<code>pam_get_item</code>	<code>pam_get_user</code>
<code>pam_getenv</code>	<code>pam_getenvlist</code>
<code>pam_open_session</code>	<code>pam_putenv</code>
<code>pam_set_data</code>	<code>pam_set_item</code>
<code>pam_setcred</code>	<code>pam_start</code>
<code>pam_strerror</code>	

FILES `/usr/lib/libpam.so.1`
 shared object

`/etc/pam.conf`
 configuration file

`/usr/lib/security/pam_dial_auth.so.1`
 authentication management PAM module for dialups

`/usr/lib/security/pam_rhosts_auth.so.1`
 authentication management PAM modules that use `ruserok()`

`/usr/lib/security/pam_sample.so.1`
 sample PAM module

`/usr/lib/security/pam_unix.so.1`
 authentication, account, session and password management PAM module

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl

libpam(3LIB)

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT Level	MT-Safe with exceptions

SEE ALSO pvs(1), intro(3), pam(3PAM), pam.conf(4), attributes(5), pam_authtok_check(5), pam_authtok_get(5), pam_authtok_store(5), pam_dial_auth(5), pam_dhkeys(5), pam_passwd_auth(5), pam_rhosts_auth(5), pam_sample(5), pam_unix(5), pam_unix_account(5), pam_unix_auth(5), pam_unix_session(5)

NOTES The functions in libpam are MT-Safe only if each thread within the multithreaded application uses its own PAM handle.

The pam_unix(5) module might not be supported in a future release. Similar functionality is provided by pam_authtok_check(5), pam_authtok_get(5), pam_authtok_store(5), pam_dhkeys(5), pam_passwd_auth(5), pam_unix_account(5), pam_unix_auth(5), and pam_unix_session(5).

NAME libpanel – panels library

SYNOPSIS `cc [flag . . .] file . . . -lpanel [library . . .]`

DESCRIPTION Functions in this library provide panels using `libcurses(3LIB)` routines.

INTERFACES The shared object `libpanel.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

<code>bottom_panel</code>	<code>del_panel</code>
<code>hide_panel</code>	<code>move_panel</code>
<code>new_panel</code>	<code>panel_above</code>
<code>panel_below</code>	<code>panel_hidden</code>
<code>panel_userptr</code>	<code>panel_window</code>
<code>replace_panel</code>	<code>set_panel_userptr</code>
<code>show_panel</code>	<code>top_panel</code>
<code>update_panels</code>	

FILES

<code>/usr/lib/libpanel.a</code>	archive library
<code>/usr/lib/libpanel.so.1</code>	shared object
<code>/usr/lib/64/libpanel.so.1</code>	64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl, SUNWarc (32-bit) SUNWcslx (64-bit)
MT-Level	Unsafe

SEE ALSO `intro(3)`, `libcurses(3LIB)`, `attributes(5)`

libpctx(3LIB)

NAME	libpctx – process context library						
SYNOPSIS	<code>cc [<i>flag</i> . . .] <i>file</i> . . . -lpctx [<i>library</i> . . .]</code>						
DESCRIPTION	<p>Functions in this library provide a simple means to access the underlying facilities of <code>proc(4)</code> to allow a controlling process to manipulate the state of a controlled process.</p> <p>This library is primarily for use in conjunction with the <code>libcpc(3LIB)</code> library. Used together, these libraries allow developers to construct tools that can manipulate CPU performance counters in other processes. The <code>cputrack(1)</code> utility is an example of such a tool.</p>						
INTERFACES	<p>The shared object <code>libpctx.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.</p> <table><tr><td><code>pctx_capture</code></td><td><code>pctx_create</code></td></tr><tr><td><code>pctx_release</code></td><td><code>pctx_run</code></td></tr><tr><td><code>pctx_set_events</code></td><td></td></tr></table>	<code>pctx_capture</code>	<code>pctx_create</code>	<code>pctx_release</code>	<code>pctx_run</code>	<code>pctx_set_events</code>	
<code>pctx_capture</code>	<code>pctx_create</code>						
<code>pctx_release</code>	<code>pctx_run</code>						
<code>pctx_set_events</code>							
FILES	<table><tr><td><code>/usr/lib/libpctx.so.1</code></td><td>shared object</td></tr><tr><td><code>/usr/lib/64/libpctx.so.1</code></td><td>64-bit shared object</td></tr></table>	<code>/usr/lib/libpctx.so.1</code>	shared object	<code>/usr/lib/64/libpctx.so.1</code>	64-bit shared object		
<code>/usr/lib/libpctx.so.1</code>	shared object						
<code>/usr/lib/64/libpctx.so.1</code>	64-bit shared object						
ATTRIBUTES	See <code>attributes(5)</code> for descriptions of the following attributes:						
	<table border="1"><thead><tr><th>ATTRIBUTE TYPE</th><th>ATTRIBUTE VALUE</th></tr></thead><tbody><tr><td>Availability</td><td>SUNWcpcu (32-bit) SUNWpcux (64-bit)</td></tr><tr><td>MT-Level</td><td>Safe</td></tr></tbody></table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	Availability	SUNWcpcu (32-bit) SUNWpcux (64-bit)	MT-Level	Safe
ATTRIBUTE TYPE	ATTRIBUTE VALUE						
Availability	SUNWcpcu (32-bit) SUNWpcux (64-bit)						
MT-Level	Safe						
SEE ALSO	<code>cputrack(1)</code> , <code>intro(3)</code> , <code>cpc(3CPC)</code> , <code>libcpc(3LIB)</code> , <code>proc(4)</code> , <code>attributes(5)</code>						

- NAME** libpicl – PICL library
- SYNOPSIS** `cc [flag . . .] file . . . -lpicl [library . . .]
#include <picl.h>`
- DESCRIPTION** Functions in this library are used to interface with the PICL daemon to access information from the PICL tree.
- INTERFACES** The shared object `libpicl.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

<code>picl_get_first_prop</code>	<code>picl_get_next_by_col</code>
<code>picl_get_next_by_row</code>	<code>picl_get_next_prop</code>
<code>picl_get_prop_by_name</code>	<code>picl_get_propinfo</code>
<code>picl_get_propinfo_by_name</code>	<code>picl_get_propval</code>
<code>picl_get_propval_by_name</code>	<code>picl_get_root</code>
<code>picl_initialize</code>	<code>picl_set_propval</code>
<code>picl_set_propval_by_name</code>	<code>picl_shutdown</code>
<code>picl_strerror</code>	<code>picl_wait</code>
<code>picl_walk_tree_by_class</code>	

- FILES** `/usr/lib/libpicl.so.1`
shared object
- `/usr/lib/64/libpicl.so.1`
64-bit shared object
- ATTRIBUTES** See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWpiclu (32-bit) SUNWpiclx (64-bit)
Interface Stability	Evolving
MT-Level	MT-Safe

- SEE ALSO** `pvs(1)`, `intro(3)`, `libpicl(3PICL)`, `attributes(5)`

libpicltree(3LIB)

- NAME** libpicltree – PICL plug-in library
- SYNOPSIS** `cc [flag . . .] file . . . -lpicltree [library . . .]`
`#include <picltree.h>`
- DESCRIPTION** Functions in this library are used to by PICL plug-in modules to register with the PICL daemon and to publish information in the PICL tree.
- INTERFACES** The shared object `libpicltree.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

<code>picld_plugin_register</code>	<code>ptree_add_node</code>
<code>ptree_add_prop</code>	<code>ptree_add_row_to_table</code>
<code>ptree_create_and_add_node</code>	<code>ptree_create_and_add_prop</code>
<code>ptree_create_node</code>	<code>ptree_create_prop</code>
<code>ptree_create_table</code>	<code>ptree_delete_node</code>
<code>ptree_delete_prop</code>	<code>ptree_destroy_node</code>
<code>ptree_destroy_prop</code>	<code>ptree_find_node</code>
<code>ptree_get_first_prop</code>	<code>ptree_get_next_by_col</code>
<code>ptree_get_next_by_row</code>	<code>ptree_get_next_prop</code>
<code>ptree_get_node_by_path</code>	<code>ptree_get_prop_by_name</code>
<code>ptree_get_propinfo</code>	<code>ptree_get_propval</code>
<code>ptree_get_propval_by_name</code>	<code>ptree_get_root</code>
<code>ptree_init_propinfo</code>	<code>ptree_post_event</code>
<code>ptree_register_handler</code>	<code>ptree_unregister_handler</code>
<code>ptree_update_propval</code>	<code>ptree_update_propval_by_name</code>
<code>ptree_walk_tree_by_class</code>	

- FILES** `/usr/lib/libpicltree.so.1`
 shared object
- ATTRIBUTES** See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWpiclu
Interface Stability	Evolving

libpicltree(3LIB)

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT-Level	MT-Safe

SEE ALSO pvs(1), intro(3), libpicltree(3PICLTREE), attributes(5)

libplot(3LIB)

NAME	libplot, lib300, lib300s, lib4014, lib450, libvt0 – graphics interface libraries																				
SYNOPSIS	<pre>cc [<i>flag</i> . . .] <i>file</i> . . . -lplot [<i>library</i> . . .] #include <plot.h></pre>																				
DESCRIPTION	Functions in this library generate graphics output.																				
INTERFACES	The shared object <code>libplot.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.																				
	<table><tr><td><code>arc</code></td><td><code>box</code></td><td><code>circle</code></td><td><code>closepl</code></td><td><code>closevt</code></td></tr><tr><td><code>cont</code></td><td><code>dot</code></td><td><code>erase</code></td><td><code>label</code></td><td><code>line</code></td></tr><tr><td><code>linemod</code></td><td><code>move</code></td><td><code>openpl</code></td><td><code>openvt</code></td><td><code>point</code></td></tr><tr><td><code>space</code></td><td></td><td></td><td></td><td></td></tr></table>	<code>arc</code>	<code>box</code>	<code>circle</code>	<code>closepl</code>	<code>closevt</code>	<code>cont</code>	<code>dot</code>	<code>erase</code>	<code>label</code>	<code>line</code>	<code>linemod</code>	<code>move</code>	<code>openpl</code>	<code>openvt</code>	<code>point</code>	<code>space</code>				
<code>arc</code>	<code>box</code>	<code>circle</code>	<code>closepl</code>	<code>closevt</code>																	
<code>cont</code>	<code>dot</code>	<code>erase</code>	<code>label</code>	<code>line</code>																	
<code>linemod</code>	<code>move</code>	<code>openpl</code>	<code>openvt</code>	<code>point</code>																	
<code>space</code>																					
FILES	<pre>/usr/lib/libplot.a archive library /usr/lib/libplot.so.1 shared object /usr/lib/sparcv9/libplot.so.1 64-bit shared object /usr/lib/lib300.a archive library /usr/lib/lib300.so.1 shared object /usr/lib/sparcv9/lib300.so.1 64-bit shared object /usr/lib/lib300s.a archive library /usr/lib/lib300s.so.1 shared object /usr/lib/sparcv9/lib300s.so.1 64-bit shared object /usr/lib/lib4014.a archive library /usr/lib/lib4014.so.1 shared object /usr/lib/sparcv9/lib4014.so.1 64-bit shared object</pre>																				


```

/usr/lib/lib450.a
  archive library
/usr/lib/lib450.so.1
  shared object
/usr/lib/sparcv9/lib450.so.1
  64-bit shared object
/usr/lib/libvt0.a
  archive library
/usr/lib/libvt0.so.1
  shared object
/usr/lib/sparcv9/libvt0.so.1
  64-bit shared object

```

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl, SUNWarc (32-bit) SUNWcslx (64-bit)
MT-Level	Unsafe

SEE ALSO `pvs(1)`, `intro(3)`, `attributes(5)`

libpool(3LIB)

NAME	libpool – pool configuration manipulation library												
SYNOPSIS	<pre>cc [flag...] file... [library...] #include <pool.h></pre>												
DESCRIPTION	<p>The functions in this library define the interface for reading and writing resource pools configuration files, as well as that for committing an existing configuration to becoming the running OS configuration (with respect to partitioning subsystems). The <code><pool.h></code> header provides type and function declarations for all library services.</p> <p>The resource pools facility brings together process-bindable resources into a common abstraction called a pool. Processor sets and other entities can be configured, grouped, and labelled in a persistent fashion such that workload components can be associated with a subset of a system's total resources. The <code>libpool</code> library provides a C language API for accessing this functionality, while <code>pooladm(1M)</code>, <code>poolbind(1M)</code>, and <code>poolcfg(1M)</code> make this facility available through command invocations from a shell. Each of those manual pages describes aspects of the pools facility; this page describes the properties available to the various entities managed within the pools facility. These entities include the system, pools, and the <code>pset</code> resources for processor sets.</p> <p>Each active entity within the resource pools framework can have an arbitrary collection of named, typed properties associated with it. Properties supported by the pools framework are listed, with descriptions, under each entity below. In general, resource properties may be one of five types: boolean, signed and unsigned integers, floating point, and string values.</p> <p>All entities and resources support a string property for commenting purposes; this property is available for use by management applications to record descriptions and other administrator oriented data. The comment field is not used by the default pools commands, except when a configuration is initiated by the <code>poolcfg</code> utility, in which case an informative message is placed in the <code>system.comment</code> property for that configuration.</p>												
System	<table><thead><tr><th>Property name</th><th>Type</th><th>Description</th></tr></thead><tbody><tr><td><code>system.bind-default</code></td><td>boolean</td><td>If specified pool not found, bind to pool with 'pool.default' property set to true.</td></tr><tr><td><code>system.comment</code></td><td>string</td><td>User description of system.</td></tr><tr><td><code>system.version</code></td><td>int</td><td><code>libpool</code> version required to manipulate this configuration.</td></tr></tbody></table> <p>The <code>system.bind-default</code> and <code>system.comment</code> properties are writable; the <code>system.version</code> property is not.</p>	Property name	Type	Description	<code>system.bind-default</code>	boolean	If specified pool not found, bind to pool with 'pool.default' property set to true.	<code>system.comment</code>	string	User description of system.	<code>system.version</code>	int	<code>libpool</code> version required to manipulate this configuration.
Property name	Type	Description											
<code>system.bind-default</code>	boolean	If specified pool not found, bind to pool with 'pool.default' property set to true.											
<code>system.comment</code>	string	User description of system.											
<code>system.version</code>	int	<code>libpool</code> version required to manipulate this configuration.											

Pools

Property name	Type	Description
<code>pool.active</code>	boolean	Mark this pool as active, if true.
<code>pool.comment</code>	string	User description of pool.
<code>pool.default</code>	boolean	Mark this pool as the default pool, if true; see <code>system.bind-default</code> property.
<code>pool.importance</code>	int	Relative importance of this pool; for possible resource dispute resolution.
<code>pool.name</code>	string	User name for pool; used by <code>setproject(3PROJECT)</code> as value for 'project.pool' project attribute in <code>project(4)</code> database.
<code>pool.scheduler</code>	string	Scheduler class to which consumers of this pool will be bound. This property is optional and if not specified, the scheduler bindings for consumers of this pool are not affected.

All of the above listed properties are writable.

Processor Sets

Property name	Type	Description
<code>pset.comment</code>	string	User description of resource.
<code>pset.default</code>	boolean	Marks default processor set.
<code>pset.escapable</code>	boolean	Represents whether <code>PSET_NOESCAPE</code> is set for this pset (see <code>pset_setattr(2)</code>)
<code>pset.load</code>	uint	The load for this processor set.
<code>pset.max</code>	uint	Maximum number of CPUs permitted in this processor set.
<code>pset.min</code>	uint	Minimum number of CPUs permitted in this processor set.
<code>pset.name</code>	string	User name for resource.
<code>pset.size</code>	uint	Current number of CPUs in this processor set.
<code>pset.sys_id</code>	int	System-assigned processor set ID.
<code>pset.type</code>	string	Names resource type; value for all processor sets is <code>pset</code> .
<code>pset.units</code>	string	Identifies meaning of size-related properties; value for all processor sets is <code>population</code> .

libpool(3LIB)

The `pset.comment`, `pset.default`, `pset.escapable`, `pset.max`, `pset.min`, `pset.min`, and `pset.name` properties are writable; the `pset.load`, `pset.size`, `pset.sys_id`, `pset.type`, and `pset.units` properties are not.

INTERFACES

The shared object `libpool.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

<code>pool_associate</code>	<code>pool_component_info</code>
<code>pool_component_to_elem</code>	<code>pool_conf_alloc</code>
<code>pool_conf_close</code>	<code>pool_conf_commit</code>
<code>pool_conf_export</code>	<code>pool_conf_free</code>
<code>pool_conf_info</code>	<code>pool_conf_location</code>
<code>pool_conf_open</code>	<code>pool_conf_remove</code>
<code>pool_conf_rollback</code>	<code>pool_conf_status</code>
<code>pool_conf_to_elem</code>	<code>pool_conf_validate</code>
<code>pool_create</code>	<code>pool_destroy</code>
<code>pool_dissociate</code>	<code>pool_dynamic_location</code>
<code>pool_error</code>	<code>pool_get_binding</code>
<code>pool_get_owning_resource</code>	<code>pool_get_pool</code>
<code>pool_get_property</code>	<code>pool_get_resource</code>
<code>pool_get_resource_binding</code>	<code>pool_info</code>
<code>pool_put_property</code>	<code>pool_query_components</code>
<code>pool_query_pool_resources</code>	<code>pool_query_pools</code>
<code>pool_query_resource_components</code>	<code>pool_query_resources</code>
<code>pool_resource_create</code>	<code>pool_resource_destroy</code>
<code>pool_resource_info</code>	<code>pool_resource_to_elem</code>
<code>pool_resource_transfer</code>	<code>pool_resource_xtransfer</code>
<code>pool_rm_property</code>	<code>pool_set_binding</code>
<code>pool_static_location</code>	<code>pool_strerror</code>
<code>pool_to_elem</code>	<code>pool_value_alloc</code>
<code>pool_value_free</code>	<code>pool_value_get_bool</code>
<code>pool_value_get_double</code>	<code>pool_value_get_int64</code>

pool_value_get_name	pool_value_get_string
pool_value_get_type	pool_value_get_uint64
pool_value_set_bool	pool_value_set_double
pool_value_set_int64	pool_value_set_name
pool_value_set_string	pool_value_set_uint64
pool_version	pool_walk_components
pool_walk_pools	pool_walk_properties
pool_walk_resources	

FILES /usr/lib/libpool.so.1
shared object

/usr/lib/sparcv9/libpool.so.1
64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWpool (32-bit) SUNWpoolx (64-bit)
CSI	Enabled
Interface Stability	Evolving
MT-Level	Unsafe

SEE ALSO intro(3), pool_component_info(3POOL), pool_conf_open(3POOL), pool_conf_to_elem(3POOL), pool_create(3POOL), pool_error(3POOL), pool_get_binding(3POOL), pool_get_property(3POOL), pool_get_resource(3POOL), pool_resource_create(3POOL), pool_value_alloc(3POOL), pool_walk_pools(3POOL), attributes(5)

NOTES Functions in libpool are unsafe for use in multithreaded applications where multiple configurations are being simultaneously manipulated by the application, due to shared state in the backing repository facility.

libproject(3LIB)

NAME	libproject – project database access library												
SYNOPSIS	<pre>cc [<i>flag</i> . . .] <i>file</i> . . . -lproject [<i>library</i> . . .] #include <project.h></pre>												
DESCRIPTION	Functions in this library provide various interfaces to extract data from the project(4) database. The header provides structure and function declarations for all library interfaces.												
INTERFACES	The shared object libproject.so.1 provides the public interfaces defined below. See intro(3) for additional information on shared object interfaces. <table><tr><td>endprojent</td><td>fgetprojent</td></tr><tr><td>getdefaultproj</td><td>getprojbyid</td></tr><tr><td>getprojbyname</td><td>getprojent</td></tr><tr><td>getprojidbyname</td><td>inproj</td></tr><tr><td>project_walk</td><td>setproject</td></tr><tr><td>setprojent</td><td></td></tr></table>	endprojent	fgetprojent	getdefaultproj	getprojbyid	getprojbyname	getprojent	getprojidbyname	inproj	project_walk	setproject	setprojent	
endprojent	fgetprojent												
getdefaultproj	getprojbyid												
getprojbyname	getprojent												
getprojidbyname	inproj												
project_walk	setproject												
setprojent													
FILES	<pre>/usr/lib/libproject.so.1 shared object /usr/lib/64/libproject.so.1 64-bit shared object</pre>												
ATTRIBUTES	See attributes(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>Availability</td><td>SUNWcsl (32-bit) SUNWcslx (64-bit)</td></tr><tr><td>Interface Stability</td><td>Evolving</td></tr><tr><td>MT-Level</td><td>Safe</td></tr></tbody></table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)	Interface Stability	Evolving	MT-Level	Safe				
ATTRIBUTE TYPE	ATTRIBUTE VALUE												
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)												
Interface Stability	Evolving												
MT-Level	Safe												
SEE ALSO	pvs(1), intro(3), getprojent(3PROJECT), project(4), attributes(5), standards(5)												

NAME	libpthread – POSIX threads library	
SYNOPSIS	cc -mt [<i>flag</i> . . .] <i>file</i> . . . -lpthread [-lrt <i>library</i> . . .]	
DESCRIPTION	Functions in this library provide routines that provide POSIX threading support. See standards(5). This library is implemented as a filter on the threads library (see libthread(3LIB)).	
INTERFACES	The shared object libpthread.so.1 provides the public interfaces defined below. See intro(3) for additional information on shared object interfaces.	
	__pthread_cleanup_pop	__pthread_cleanup_push
	pthread_attr_destroy	pthread_attr_getdetachstate
	pthread_attr_getguardsize	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_setguardsize
	pthread_attr_setinheritsched	pthread_attr_setschedparam
	pthread_attr_setschedpolicy	pthread_attr_setscope
	pthread_attr_setstackaddr	pthread_attr_setstacksize
	pthread_cancel	pthread_cond_broadcast
	pthread_cond_destroy	pthread_cond_init
	pthread_cond_reltimedwait_np	pthread_cond_signal
	pthread_cond_timedwait	pthread_cond_wait
	pthread_condattr_destroy	pthread_condattr_getpshared
	pthread_condattr_init	pthread_condattr_setpshared
	pthread_create	pthread_detach
	pthread_equal	pthread_exit
	pthread_getconcurrency	pthread_getschedparam
	pthread_getspecific	pthread_join
	pthread_key_create	pthread_key_delete
	pthread_kill	pthread_mutex_consistent_np
	pthread_mutex_destroy	pthread_mutex_getprioceiling

libpthread(3LIB)

pthread_mutex_init	pthread_mutex_lock
pthread_mutex_setprioceiling	pthread_mutex_trylock
pthread_mutex_unlock	pthread_mutexattr_destroy
pthread_mutexattr_getprioceiling	pthread_mutexattr_getprotocol
pthread_mutexattr_getpshared	pthread_mutexattr_getrobust_np
pthread_mutexattr_gettype	pthread_mutexattr_init
pthread_mutexattr_setprioceiling	pthread_mutexattr_setprotocol
pthread_mutexattr_setpshared	pthread_mutexattr_setrobust_np
pthread_mutexattr_settype	pthread_once
pthread_rwlock_destroy	pthread_rwlock_init
pthread_rwlock_rdlock	pthread_rwlock_tryrdlock
pthread_rwlock_trywrlock	pthread_rwlock_unlock
pthread_rwlock_wrlock	pthread_rwlockattr_destroy
pthread_rwlockattr_getpshared	pthread_rwlockattr_init
pthread_rwlockattr_setpshared	pthread_self
pthread_setcancelstate	pthread_setcanceltype
pthread_setconcurrency	pthread_setschedparam
pthread_setspecific	pthread_sigmask
pthread_testcancel	

FILES

/usr/lib/libpthread.so.1
a filter on libthread.so.1

/usr/lib/64/libpthread.so.1
a filter on the 64-bit version of libthread.so.1

ATTRIBUTES

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)
MT-Level	Safe

libpthread(3LIB)

SEE ALSO pvs(1), intro(2), intro(3), libpthread(3LIB), libthread(3LIB),
libthread_db(3LIB), libthread_db(3THR), threads(3THR), attributes(5)

librac(3LIB)

NAME librac – remote asynchronous calls library

SYNOPSIS `cc [flag . . .] file . . . -lrac -lnsl [library . . .]`
`#include <rpc/rpc.h>`
`#include <rpc/rac.h>`

DESCRIPTION Functions in this library provide a remote asynchronous call interface to the RPC library.

INTERFACES The shared object `librac.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

<code>clnt_create</code>	<code>clnt_create_vers</code>
<code>clnt_dg_create</code>	<code>clnt_tli_create</code>
<code>clnt_tp_create</code>	<code>clnt_vc_create</code>
<code>rac_drop</code>	<code>rac_poll</code>
<code>rac_recv</code>	<code>rac_send</code>
<code>rac_senderr</code>	<code>rpcb_getaddr</code>
<code>rpcb_getmaps</code>	<code>rpcb_gettime</code>
<code>rpcb_rmtcall</code>	<code>rpcb_set</code>
<code>rpcb_taddr2uaddr</code>	<code>rpcb_uaddr2taddr</code>
<code>rpcb_unset</code>	<code>xdrrec_create</code>
<code>xdrrec_endofrecord</code>	<code>xdrrec_eof</code>
<code>xdrrec_readbytes</code>	<code>xdrrec_skiprecord</code>

FILES

<code>/usr/lib/librac.a</code>	archive library
<code>/usr/lib/librac.so.1</code>	shared object
<code>/usr/lib/64/librac.so.1</code>	64-bit shared object file

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl, SUNWarc (32-bit) SUNWcslx (64-bit)
MT-Level	Unsafe

SEE ALSO `pvs(1)`, `intro(3)`, `rpc_rac(3RAC)`, `attributes(5)`

NAME	libresolv – resolver library																																		
SYNOPSIS	<pre>cc [<i>flag</i> . . .] <i>file</i> . . . -lresolv -lsocket -lnsl [<i>library</i> . . .] #include <sys/types.h> #include <netinet/in.h> #include <arpa/nameser.h> #include <resolv.h> #include <netdb.h></pre>																																		
DESCRIPTION	Functions in this library provide for creating, sending, and interpreting packets to the Internet domain name servers.																																		
INTERFACES	<p>The shared object <code>libresolv.so.2</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.</p> <table border="0"> <tr><td><code>__dn_skipname</code></td><td><code>__fp_query</code></td></tr> <tr><td><code>__hostalias</code></td><td><code>__p_cdname</code></td></tr> <tr><td><code>__p_class</code></td><td><code>__p_query</code></td></tr> <tr><td><code>__p_time</code></td><td><code>__p_type</code></td></tr> <tr><td><code>__putlong</code></td><td><code>_getlong</code></td></tr> <tr><td><code>_getshort</code></td><td><code>_res</code></td></tr> <tr><td><code>dn_comp</code></td><td><code>dn_expand</code></td></tr> <tr><td><code>fp_resstat</code></td><td><code>h_errno</code></td></tr> <tr><td><code>herror</code></td><td><code>hstrerror</code></td></tr> <tr><td><code>res_hostalias</code></td><td><code>res_init</code></td></tr> <tr><td><code>res_mkquery</code></td><td><code>res_nclose</code></td></tr> <tr><td><code>res_ninit</code></td><td><code>res_nmkquery</code></td></tr> <tr><td><code>res_nquery</code></td><td><code>res_nquerydomain</code></td></tr> <tr><td><code>res_nsearch</code></td><td><code>res_nsend</code></td></tr> <tr><td><code>res_nsendsigned</code></td><td><code>res_query</code></td></tr> <tr><td><code>res_querydomain</code></td><td><code>res_search</code></td></tr> <tr><td><code>res_send</code></td><td><code>res_update</code></td></tr> </table> <p>Programs are expected to use the aliases defined in <code><resolv.h></code> rather than calling the "<code>__</code>" prefixed procedures, as indicated in the following table. Use of the routines in the first column is discouraged.</p>	<code>__dn_skipname</code>	<code>__fp_query</code>	<code>__hostalias</code>	<code>__p_cdname</code>	<code>__p_class</code>	<code>__p_query</code>	<code>__p_time</code>	<code>__p_type</code>	<code>__putlong</code>	<code>_getlong</code>	<code>_getshort</code>	<code>_res</code>	<code>dn_comp</code>	<code>dn_expand</code>	<code>fp_resstat</code>	<code>h_errno</code>	<code>herror</code>	<code>hstrerror</code>	<code>res_hostalias</code>	<code>res_init</code>	<code>res_mkquery</code>	<code>res_nclose</code>	<code>res_ninit</code>	<code>res_nmkquery</code>	<code>res_nquery</code>	<code>res_nquerydomain</code>	<code>res_nsearch</code>	<code>res_nsend</code>	<code>res_nsendsigned</code>	<code>res_query</code>	<code>res_querydomain</code>	<code>res_search</code>	<code>res_send</code>	<code>res_update</code>
<code>__dn_skipname</code>	<code>__fp_query</code>																																		
<code>__hostalias</code>	<code>__p_cdname</code>																																		
<code>__p_class</code>	<code>__p_query</code>																																		
<code>__p_time</code>	<code>__p_type</code>																																		
<code>__putlong</code>	<code>_getlong</code>																																		
<code>_getshort</code>	<code>_res</code>																																		
<code>dn_comp</code>	<code>dn_expand</code>																																		
<code>fp_resstat</code>	<code>h_errno</code>																																		
<code>herror</code>	<code>hstrerror</code>																																		
<code>res_hostalias</code>	<code>res_init</code>																																		
<code>res_mkquery</code>	<code>res_nclose</code>																																		
<code>res_ninit</code>	<code>res_nmkquery</code>																																		
<code>res_nquery</code>	<code>res_nquerydomain</code>																																		
<code>res_nsearch</code>	<code>res_nsend</code>																																		
<code>res_nsendsigned</code>	<code>res_query</code>																																		
<code>res_querydomain</code>	<code>res_search</code>																																		
<code>res_send</code>	<code>res_update</code>																																		

libresolv(3LIB)

FUNCTION REFERENCED	ALIAS TO USE
__dn_skipname	dn_skipname
__fp_query	fp_query
__putlong	putlong
__p_cdname	p_cdname
__p_class	p_class
__p_time	p_time
__p_type	p_type

FILES /usr/lib/libresolv.so.1
 shared object for backward compatibility only
 /usr/lib/64/libresolv.so.1
 64-bit shared object for backward compatibility only
 /usr/lib/libresolv.so.2
 shared object
 /usr/lib/64/libresolv.so.2
 64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)
MT-Level	See resolver(3RESOLV)
Interface Stability	BIND 8.2.4

SEE ALSO pvs(1), intro(3), resolver(3RESOLV), attributes(5)

NAME librpcsoc – obsolete RPC library

SYNOPSIS `cc [flag . . .] -I /usr/ucbinclude file . . . -L /usr/libucb \`
`-R /usr/libucb -lrpcsoc [library . . .]`
`#include <rpc/rpc.h>`

DESCRIPTION Functions in this library implement socket based RPC calls (using socket calls, not TLI). Applications that require this library should link it before `libnsl`, which implements the same calls over TLI .

This library is provided for compatibility only; new applications should not link in this library.

INTERFACES The shared object `librpcsoc.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

<code>clnttcp_create</code>	<code>clntudp_bufcreate</code>
<code>clntudp_create</code>	<code>get_myaddress</code>
<code>getrpcport</code>	<code>rtime</code>
<code>svcfld_create</code>	<code>svctcp_create</code>
<code>svcurdp_bufcreate</code>	<code>svcurdp_create</code>
<code>svcurdp_enablecache</code>	

FILES `/usr/ucblib/librpcsoc.so.1`
 shared object

`/usr/ucblib/64/librpcsoc.so.1`
 64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWscpu (32-bit) SUNWscpux (64-bit)
MT-Level	Unsafe

SEE ALSO `pvs(1)`, `intro(3)`, `rpc_soc(3NSL)`, `libnsl(3LIB)`, `attributes(5)`

librpcsvc(3LIB)

NAME	librpcsvc – RPC services library								
SYNOPSIS	<pre>cc [<i>flag</i> . . .] <i>file</i> . . . -lrpcsvc [<i>library</i> . . .] #include <rpc/rpc.h> #include <rpcsvc/rstat.h></pre>								
DESCRIPTION	Functions in this library provide RPC services. See the manual pages in Section 3RPC for the individual functions.								
INTERFACES	The shared object <code>librpcsvc.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces. <table><tr><td><code>havedisk</code></td><td><code>rnusers</code></td></tr><tr><td><code>rstat</code></td><td><code>rusers</code></td></tr><tr><td><code>rwall</code></td><td><code>xdr_statstime</code></td></tr><tr><td><code>xdr_statsvar</code></td><td><code>xdr_utmpidlearr</code></td></tr></table>	<code>havedisk</code>	<code>rnusers</code>	<code>rstat</code>	<code>rusers</code>	<code>rwall</code>	<code>xdr_statstime</code>	<code>xdr_statsvar</code>	<code>xdr_utmpidlearr</code>
<code>havedisk</code>	<code>rnusers</code>								
<code>rstat</code>	<code>rusers</code>								
<code>rwall</code>	<code>xdr_statstime</code>								
<code>xdr_statsvar</code>	<code>xdr_utmpidlearr</code>								
FILES	<pre>/usr/lib/librpcsvc.a archive library /usr/lib/librpcsvc.so.1 shared object /usr/lib/64/librpcsvc.so.1 64-bit shared object</pre>								
ATTRIBUTES	See <code>attributes(5)</code> for descriptions of the following attributes: <table border="1"><thead><tr><th>ATTRIBUTE TYPE</th><th>ATTRIBUTE VALUE</th></tr></thead><tbody><tr><td>Availability</td><td>SUNWcsl, SUNWarc (32-bit) SUNWcslx (64-bit)</td></tr><tr><td>MT-Level</td><td>Safe</td></tr></tbody></table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	Availability	SUNWcsl, SUNWarc (32-bit) SUNWcslx (64-bit)	MT-Level	Safe		
ATTRIBUTE TYPE	ATTRIBUTE VALUE								
Availability	SUNWcsl, SUNWarc (32-bit) SUNWcslx (64-bit)								
MT-Level	Safe								
SEE ALSO	<code>pvs(1)</code> , <code>intro(3)</code> , <code>rstat(3RPC)</code> , <code>attributes(5)</code>								

NAME	librsm – remote shared memory interface library																																										
SYNOPSIS	cc [<i>flag</i> . . .] <i>file</i> . . . -lrsm [<i>library</i> . . .] #include <rsmapi.h>																																										
DESCRIPTION	The functions in this library provide an interface for OS bypass messaging for applications over high-speed interconnects, including facilities to set up low-latency, high-bandwidth interprocess communication mechanisms and to perform I/O.																																										
INTERFACES	The shared object librsm.so.2 provides the public interfaces defined below. See intro(3) for additional information on shared object interfaces.																																										
	<table border="0" style="width: 100%;"> <tr> <td>rsm_create_localmemory_handle</td> <td>rsm_free_interconnect_topology</td> </tr> <tr> <td>rsm_free_localmemory_handle</td> <td>rsm_get_controller</td> </tr> <tr> <td>rsm_get_controller_attr</td> <td>rsm_get_interconnect_topology</td> </tr> <tr> <td>rsm_get_segmentid_range</td> <td>rsm_intr_signal_post</td> </tr> <tr> <td>rsm_intr_signal_wait</td> <td>rsm_memseg_export_create</td> </tr> <tr> <td>rsm_memseg_export_destroy</td> <td>rsm_memseg_export_publish</td> </tr> <tr> <td>rsm_memseg_export_rebind</td> <td>rsm_memseg_export_republish</td> </tr> <tr> <td>rsm_memseg_export_unpublish</td> <td>rsm_memseg_get_pollfd</td> </tr> <tr> <td>rsm_memseg_import_close_barrier</td> <td>rsm_memseg_import_connect</td> </tr> <tr> <td>rsm_memseg_import_destroy_barrier</td> <td>rsm_memseg_import_disconnect</td> </tr> <tr> <td>rsm_memseg_import_get</td> <td>rsm_memseg_import_get16</td> </tr> <tr> <td>rsm_memseg_import_get32</td> <td>rsm_memseg_import_get64</td> </tr> <tr> <td>rsm_memseg_import_get8</td> <td>rsm_memseg_import_get_mode</td> </tr> <tr> <td>rsm_memseg_import_getv</td> <td>rsm_memseg_import_init_barrier</td> </tr> <tr> <td>rsm_memseg_import_map</td> <td>rsm_memseg_import_open_barrier</td> </tr> <tr> <td>rsm_memseg_import_order_barrier</td> <td>rsm_memseg_import_put</td> </tr> <tr> <td>rsm_memseg_import_put16</td> <td>rsm_memseg_import_put32</td> </tr> <tr> <td>rsm_memseg_import_put64</td> <td>rsm_memseg_import_put8</td> </tr> <tr> <td>rsm_memseg_import_putv</td> <td>rsm_memseg_import_set_mode</td> </tr> <tr> <td>rsm_memseg_import_unmap</td> <td>rsm_memseg_release_pollfd</td> </tr> <tr> <td>rsm_release_controller</td> <td></td> </tr> </table>	rsm_create_localmemory_handle	rsm_free_interconnect_topology	rsm_free_localmemory_handle	rsm_get_controller	rsm_get_controller_attr	rsm_get_interconnect_topology	rsm_get_segmentid_range	rsm_intr_signal_post	rsm_intr_signal_wait	rsm_memseg_export_create	rsm_memseg_export_destroy	rsm_memseg_export_publish	rsm_memseg_export_rebind	rsm_memseg_export_republish	rsm_memseg_export_unpublish	rsm_memseg_get_pollfd	rsm_memseg_import_close_barrier	rsm_memseg_import_connect	rsm_memseg_import_destroy_barrier	rsm_memseg_import_disconnect	rsm_memseg_import_get	rsm_memseg_import_get16	rsm_memseg_import_get32	rsm_memseg_import_get64	rsm_memseg_import_get8	rsm_memseg_import_get_mode	rsm_memseg_import_getv	rsm_memseg_import_init_barrier	rsm_memseg_import_map	rsm_memseg_import_open_barrier	rsm_memseg_import_order_barrier	rsm_memseg_import_put	rsm_memseg_import_put16	rsm_memseg_import_put32	rsm_memseg_import_put64	rsm_memseg_import_put8	rsm_memseg_import_putv	rsm_memseg_import_set_mode	rsm_memseg_import_unmap	rsm_memseg_release_pollfd	rsm_release_controller	
rsm_create_localmemory_handle	rsm_free_interconnect_topology																																										
rsm_free_localmemory_handle	rsm_get_controller																																										
rsm_get_controller_attr	rsm_get_interconnect_topology																																										
rsm_get_segmentid_range	rsm_intr_signal_post																																										
rsm_intr_signal_wait	rsm_memseg_export_create																																										
rsm_memseg_export_destroy	rsm_memseg_export_publish																																										
rsm_memseg_export_rebind	rsm_memseg_export_republish																																										
rsm_memseg_export_unpublish	rsm_memseg_get_pollfd																																										
rsm_memseg_import_close_barrier	rsm_memseg_import_connect																																										
rsm_memseg_import_destroy_barrier	rsm_memseg_import_disconnect																																										
rsm_memseg_import_get	rsm_memseg_import_get16																																										
rsm_memseg_import_get32	rsm_memseg_import_get64																																										
rsm_memseg_import_get8	rsm_memseg_import_get_mode																																										
rsm_memseg_import_getv	rsm_memseg_import_init_barrier																																										
rsm_memseg_import_map	rsm_memseg_import_open_barrier																																										
rsm_memseg_import_order_barrier	rsm_memseg_import_put																																										
rsm_memseg_import_put16	rsm_memseg_import_put32																																										
rsm_memseg_import_put64	rsm_memseg_import_put8																																										
rsm_memseg_import_putv	rsm_memseg_import_set_mode																																										
rsm_memseg_import_unmap	rsm_memseg_release_pollfd																																										
rsm_release_controller																																											
FILES	/usr/lib/librsm.so.2 shared object																																										

librsm(3LIB)

/usr/lib/64/librsm.so.2
64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWrsm (32-bit) SUNWrsmx (64-bit)
Interface Stability	Evolving
MT-Level	Safe

SEE ALSO `intro(2)`, `intro(3)`, `attributes(5)`

NAME	librt, libposix4 – POSIX.1b Realtime Extensions library																																		
SYNOPSIS	<pre>cc [<i>flag</i> . . .] <i>file</i> . . . -lrt [<i>library</i> . . .] cc [<i>flag</i> . . .] <i>file</i> . . . -lposix4 [<i>library</i> . . .]</pre>																																		
DESCRIPTION	<p>Functions in this library provide most of the interfaces specified by the POSIX.1b Realtime Extension. See <code>standards(5)</code>. 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 <code>libc(3LIB)</code>.</p> <p>See the man pages for the individual interfaces in section 3RT for information on required headers.</p> <p>The name <code>libposix4</code> is maintained for backward compatibility and should be avoided. <code>librt</code> is the preferred name for this library.</p>																																		
INTERFACES	<p>The shared objects <code>librt.so.1</code> and <code>libposix4.so.1</code> provide the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.</p> <table border="0" style="width: 100%;"> <tr><td><code>aio_cancel</code></td><td><code>aio_error</code></td></tr> <tr><td><code>aio_fsync</code></td><td><code>aio_read</code></td></tr> <tr><td><code>aio_return</code></td><td><code>aio_suspend</code></td></tr> <tr><td><code>aio_waitn</code></td><td><code>aio_write</code></td></tr> <tr><td><code>clock_getres</code></td><td><code>clock_gettime</code></td></tr> <tr><td><code>clock_settime</code></td><td><code>close</code></td></tr> <tr><td><code>fdatasync</code></td><td><code>fork</code></td></tr> <tr><td><code>lio_listio</code></td><td><code>mq_close</code></td></tr> <tr><td><code>mq_getattr</code></td><td><code>mq_notify</code></td></tr> <tr><td><code>mq_open</code></td><td><code>mq_receive</code></td></tr> <tr><td><code>mq_send</code></td><td><code>mq_setattr</code></td></tr> <tr><td><code>mq_unlink</code></td><td><code>nanosleep</code></td></tr> <tr><td><code>sched_get_priority_max</code></td><td><code>sched_get_priority_min</code></td></tr> <tr><td><code>sched_getparam</code></td><td><code>sched_getscheduler</code></td></tr> <tr><td><code>sched_rr_get_interval</code></td><td><code>sched_setparam</code></td></tr> <tr><td><code>sched_setscheduler</code></td><td><code>sched_yield</code></td></tr> <tr><td><code>sem_close</code></td><td><code>sem_destroy</code></td></tr> </table>	<code>aio_cancel</code>	<code>aio_error</code>	<code>aio_fsync</code>	<code>aio_read</code>	<code>aio_return</code>	<code>aio_suspend</code>	<code>aio_waitn</code>	<code>aio_write</code>	<code>clock_getres</code>	<code>clock_gettime</code>	<code>clock_settime</code>	<code>close</code>	<code>fdatasync</code>	<code>fork</code>	<code>lio_listio</code>	<code>mq_close</code>	<code>mq_getattr</code>	<code>mq_notify</code>	<code>mq_open</code>	<code>mq_receive</code>	<code>mq_send</code>	<code>mq_setattr</code>	<code>mq_unlink</code>	<code>nanosleep</code>	<code>sched_get_priority_max</code>	<code>sched_get_priority_min</code>	<code>sched_getparam</code>	<code>sched_getscheduler</code>	<code>sched_rr_get_interval</code>	<code>sched_setparam</code>	<code>sched_setscheduler</code>	<code>sched_yield</code>	<code>sem_close</code>	<code>sem_destroy</code>
<code>aio_cancel</code>	<code>aio_error</code>																																		
<code>aio_fsync</code>	<code>aio_read</code>																																		
<code>aio_return</code>	<code>aio_suspend</code>																																		
<code>aio_waitn</code>	<code>aio_write</code>																																		
<code>clock_getres</code>	<code>clock_gettime</code>																																		
<code>clock_settime</code>	<code>close</code>																																		
<code>fdatasync</code>	<code>fork</code>																																		
<code>lio_listio</code>	<code>mq_close</code>																																		
<code>mq_getattr</code>	<code>mq_notify</code>																																		
<code>mq_open</code>	<code>mq_receive</code>																																		
<code>mq_send</code>	<code>mq_setattr</code>																																		
<code>mq_unlink</code>	<code>nanosleep</code>																																		
<code>sched_get_priority_max</code>	<code>sched_get_priority_min</code>																																		
<code>sched_getparam</code>	<code>sched_getscheduler</code>																																		
<code>sched_rr_get_interval</code>	<code>sched_setparam</code>																																		
<code>sched_setscheduler</code>	<code>sched_yield</code>																																		
<code>sem_close</code>	<code>sem_destroy</code>																																		

librt(3LIB)

sem_getvalue	sem_init
sem_open	sem_post
sem_trywait	sem_unlink
sem_wait	shm_open
shm_unlink	sigqueue
sigtimedwait	sigwaitinfo
timer_create	timer_delete
timer_getoverrun	timer_gettime
timer_settime	

The following interfaces are unique to the 32-bit version of this library:

aio_cancel64	aio_error64
aio_fsync64	aio_read64
aio_return64	aio_suspend64
aio_write64	lio_listio64

FILES

/usr/lib/librt.so.1
shared object

/usr/lib/64/librt.so.1
64-bit shared object file

/usr/lib/libposix4.so.1
shared object

/usr/lib/64/libposix4.so.1
64-bit shared object file

ATTRIBUTES

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)
MT-Level	Safe

SEE ALSO

pvs(1), intro(3), libc(3LIB), attributes(5), standards(5)

NAME librtld_db – runtime linker debugging library

SYNOPSIS `cc [flag ...] file ... -lrtld_db [library ...]`
`#include <proc_service.h>`
`#include <rtld_db.h>`

DESCRIPTION Functions in this library are useful for building debuggers for dynamically linked programs. For a full description of these interfaces refer to the *Linker and Libraries Guide*.

INTERFACES The shared object `librtld_db.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

<code>rd_delete</code>	<code>rd_errstr</code>
<code>rd_event_addr</code>	<code>rd_event_enable</code>
<code>rd_event_getmsg</code>	<code>rd_init</code>
<code>rd_loadobj_iter</code>	<code>rd_log</code>
<code>rd_new</code>	<code>rd_objpad_enable</code>
<code>rd_plt_resolution</code>	<code>rd_reset</code>

FILES `/usr/lib/librtld_db.so.1`
shared object
`/usr/lib/64/librtld_db.so.1`
64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)
MT-Level	Safe

SEE ALSO `ld.so.1(1)`, `pvs(1)`, `intro(3)`, `rtld_db(3EXT)`, `attributes(5)`
Linker and Libraries Guide

libsec(3LIB)

NAME libsec – File Access Control List library

SYNOPSIS `cc [flag . . .] file . . . -lsec [library . . .]`
`#include <sys/acl.h>`

DESCRIPTION Functions in this library provide comparison and manipulation of File Access Control Lists.

INTERFACES The shared object `libsec.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

<code>aclcheck</code>	<code>aclfrommode</code>	<code>aclfromtext</code>
<code>aclsort</code>	<code>acltomode</code>	<code>acltotext</code>

FILES

<code>/usr/lib/libsec.a</code>	archive library
<code>/usr/lib/libsec.so.1</code>	shared object
<code>/usr/lib/64/libsec.so.1</code>	64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl, SUNWarc (32-bit) SUNWcslx (64-bit)
Interface Stability	Evolving
MT-Level	Unsafe

SEE ALSO `pvs(1)`, `intro(3)`, `attributes(5)`

NAME	libsecdb – security attributes database library	
SYNOPSIS	<pre>cc [<i>flag</i> . . .] <i>file</i> . . . -lsecdb [<i>library</i> . . .] #include <secdb.h> #include <user_attr.h> #include <prof_attr.h> #include <exec_attr.h> #include <auth_attr.h></pre>	
DESCRIPTION	Functions in this library provide routines for manipulation of security attribute databases.	
INTERFACES	The shared object <code>libsecdb.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.	
	<pre>chkauthattr endexecattr enduserattr free_authattr free_profattr free_userattr getauthnam getexecprof getprofattr getprofnam getusernam kva_match setauthattr setprofattr</pre>	<pre>endauthattr endprofattr fgetuserattr free_execattr free_proflist getauthattr getexecattr getexecuser getproflist getuserattr getuserid match_execattr setexecattr setuserattr</pre>
FILES	<pre>/usr/lib/libsecdb.so.1 /usr/lib/64/libsecdb.so.1</pre>	<pre>shared object 64-bit shared object</pre>
ATTRIBUTES	See <code>attributes(5)</code> for description of the following attributes:	

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl (32-bit)

libsecdb(3LIB)

ATTRIBUTE TYPE	ATTRIBUTE VALUE
	SUNWcslx (64-bit)
MT Level	MT-Safe

SEE ALSO [intro\(3\)](#), [attributes\(5\)](#)

NAME	libsendfile – sendfile library								
SYNOPSIS	cc [<i>flag...</i>] <i>file...</i> -lsendfile [<i>library...</i>] #include <sys/sendfile.h>								
DESCRIPTION	The functions in this library provide routines that enable files to be sent over sockets, buffers to be sent over sockets, files to be copied to files, and buffers to be copied to files.								
INTERFACES	The shared object <code>libsendfile.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.								
	<table> <tr> <td><code>sendfile</code></td> <td><code>sendfilev</code></td> </tr> </table> <p>The following interfaces are unique to the 32-bit version of this library:</p> <table> <tr> <td><code>sendfile64</code></td> <td><code>sendfilev64</code></td> </tr> </table>	<code>sendfile</code>	<code>sendfilev</code>	<code>sendfile64</code>	<code>sendfilev64</code>				
<code>sendfile</code>	<code>sendfilev</code>								
<code>sendfile64</code>	<code>sendfilev64</code>								
FILES	<table> <tr> <td><code>/usr/lib/libsendfile.so.1</code></td> <td>shared object</td> </tr> <tr> <td><code>/usr/lib/64/libsendfile.so.1</code></td> <td>64-bit shared object</td> </tr> </table>	<code>/usr/lib/libsendfile.so.1</code>	shared object	<code>/usr/lib/64/libsendfile.so.1</code>	64-bit shared object				
<code>/usr/lib/libsendfile.so.1</code>	shared object								
<code>/usr/lib/64/libsendfile.so.1</code>	64-bit shared object								
ATTRIBUTES	See <code>attributes(5)</code> for descriptions of the following attributes:								
	<table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>Availability</td> <td>SUNWcsl (32-bit) SUNWcslx (64-bit)</td> </tr> <tr> <td>Interface Stability</td> <td>Evolving</td> </tr> <tr> <td>MT-Level</td> <td>MT-Safe</td> </tr> </tbody> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)	Interface Stability	Evolving	MT-Level	MT-Safe
ATTRIBUTE TYPE	ATTRIBUTE VALUE								
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)								
Interface Stability	Evolving								
MT-Level	MT-Safe								
SEE ALSO	<code>pvs(1)</code> , <code>intro(3)</code> , <code>sendfile(3EXT)</code> , <code>sendfilev(3EXT)</code> , <code>attributes(5)</code>								

libslp(3LIB)

NAME libslp – service location protocol library

SYNOPSIS cc [*flag* . . .] *file* . . . -lslp [*library* . . .]

DESCRIPTION Functions in this library provide routines that provide the Service Location Protocol C library.

INTERFACES The shared object `libslp.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

SLPClose	SLPDelAttrs
SLPDereg	SLPEscape
SLPFindAttrs	SLPFindScopes
SLPFindSrvTypes	SLPFindSrvs
SLPFree	SLPGetProperty
SLPGetRefreshInterval	SLPOpen
SLPParseSrvURL	SLPReg
SLPSetProperty	SLPUnescape
slp_strerror	

FILES /usr/lib/libslp.so.1
shared object

/usr/lib/64/libslp.so.1
64-bit shared object file

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWslpu

SEE ALSO `pvs(1)`, `intro(2)`, `intro(3)`, `attributes(5)`

NAME libsmartcard – smartcard library

SYNOPSIS `cc [flag...] file... -lsmartcard [library...]`
`#include <smartcard/scf.h>`

DESCRIPTION Functions in this library allow an application to select a smartcard terminal, determine when cards are inserted or removed, and exchange data with the card.

INTERFACES The shared object `libsmartcard.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

<code>SCF_Card_close</code>	<code>SCF_Card_exchangeAPDU</code>
<code>SCF_Card_freeInfo</code>	<code>SCF_Card_getInfo</code>
<code>SCF_Card_lock</code>	<code>SCF_Card_reset</code>
<code>SCF_Card_unlock</code>	<code>SCF_Card_waitForCardRemoved</code>
<code>SCF_Session_close</code>	<code>SCF_Session_freeInfo</code>
<code>SCF_Session_getInfo</code>	<code>SCF_Session_getSession</code>
<code>SCF_Session_getTerminal</code>	<code>SCF_Terminal_addEventListener</code>
<code>SCF_Terminal_close</code>	<code>SCF_Terminal_freeInfo</code>
<code>SCF_Terminal_getCard</code>	<code>SCF_Terminal_getInfo</code>
<code>SCF_Terminal_removeEventListener</code>	<code>SCF_Terminal_updateEventListener</code>
<code>SCF_Terminal_waitForCardAbsent</code>	<code>SCF_Terminal_waitForCardPresent</code>
<code>SCF_strerror</code>	

FILES `/usr/lib/libsmartcard.so.1`
 shared object

`/usr/lib/64/libsmartcard.so.1`
 64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWocf (32-bit) SUNWocfx (64-bit)
Interface Stability	Evolving
MT-Level	MT-Safe

libsmartcard(3LIB)

SEE ALSO smartcard(1M), intro(3), attributes(5), smartcard(5)

NAME	libsocket – sockets library	
SYNOPSIS	cc [<i>flag</i> . . .] <i>file</i> . . . -lsocket [<i>library</i> . . .]	
DESCRIPTION	Functions in this library provide routines that provide the socket internetworking interface, primarily used with the TCP/IP protocol suite.	
INTERFACES	The shared object <code>libsocket.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.	
	<code>__xnet_bind</code>	<code>__xnet_connect</code>
	<code>__xnet_getsockopt</code>	<code>__xnet_listen</code>
	<code>__xnet_recvmsg</code>	<code>__xnet_sendmsg</code>
	<code>__xnet_sendto</code>	<code>__xnet_socket</code>
	<code>__xnet_socketpair</code>	<code>accept</code>
	<code>bind</code>	<code>bindresvport</code>
	<code>connect</code>	<code>endnetent</code>
	<code>endprotoent</code>	<code>endservent</code>
	<code>ether_aton</code>	<code>ether_hostton</code>
	<code>ether_line</code>	<code>ether_ntoa</code>
	<code>ether_ntohost</code>	<code>freeaddrinfo</code>
	<code>gai_strerror</code>	<code>getaddrinfo</code>
	<code>getnameinfo</code>	<code>getnetbyaddr</code>
	<code>getnetbyaddr_r</code>	<code>getnetbyname</code>
	<code>getnetbyname_r</code>	<code>getnetent</code>
	<code>getnetent_r</code>	<code>getpeername</code>
	<code>getprotobyname</code>	<code>getprotobyname_r</code>
	<code>getprotobynumber</code>	<code>getprotobynumber_r</code>
	<code>getprotoent</code>	<code>getprotoent_r</code>
	<code>getservbyname</code>	<code>getservbyname_r</code>
	<code>getservbyport</code>	<code>getservbyport_r</code>
	<code>getservent</code>	<code>getservent_r</code>
	<code>getsockname</code>	<code>getsockopt</code>
	<code>htonl</code>	<code>htons</code>

libsocket(3LIB)

if_freenameindex	if_indextoname
if_nameindex	if_nametoindex
in6addr_any	in6addr_loopback
inet_lnaof	inet_makeaddr
inet_network	listen
ntohl	ntohs
rcmd	rcmd_af
recv	recvfrom
recvmsg	rexec
rexec_af	rresvport
rresvport_af	ruserok
send	sendmsg
sendto	setnetent
setprotoent	setservent
setsockopt	shutdown
socket	socketpair

FILES /usr/lib/libsocket.a
 archive library

/usr/lib/libsocket.so.1
 shared object

/usr/lib/64/libsocket.so.1
 64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl, SUNWarc (32-bit) SUNWcslx (64-bit)
MT-Level	Safe

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

NAME libssagent – Sun Solstice Enterprise Agent library

SYNOPSIS `cc [flag . . .] file . . . -lssagent [library . . .]`

DESCRIPTION The libssagent library is a high level API library that is dependent on libssasnm. 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.

INTERFACES The shared object libssagent.so.1 provides the public interfaces defined below. See intro(3) for additional information on shared object interfaces.

SSAgentIsAlive	SSAGetTrapPort
SSAMain	SSARegSubagent
SSARegSubtree	SSASubagentOpen
_SSASendTrap	_SSASendTrap2
_SSASendTrap3	callItem
numCallItem	numTrapElem
trapAnyEnterpriseInfo	trapBucket
trapEnterpriseInfo	trapTableMap

FILES /usr/lib/libssagent.so.1 shared object
/usr/lib/64/libssagent.so.1 64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWsasnm
MT-Level	Unsafe

SEE ALSO intro(3), libssasnm(3LIB), attributes(5)

libssasnm(3LIB)

NAME	libssasnm – Sun Solstice Enterprise SNMP library														
SYNOPSIS	<code>cc [<i>flag</i> . . .] <i>file</i> . . . -lssasnm [<i>library</i> . . .]</code>														
DESCRIPTION	<p>The libssasnm library provides low-level SNMP API functions.</p> <ul style="list-style-type: none">■ ASN.1 serialization (encoding/decoding) module■ SNMP PDU development routines■ SNMP session module■ Low level SNMP based API functions■ Error-handling module■ Trace (debugging) module														
INTERFACES	<p>The shared object <code>libssasnm.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.</p> <table><tr><td><code>SSAOidCmp</code></td><td><code>SSAOidCpy</code></td></tr><tr><td><code>SSAOidDup</code></td><td><code>SSAOidFree</code></td></tr><tr><td><code>SSAOidInit</code></td><td><code>SSAOidNew</code></td></tr><tr><td><code>SSAOidStrToOid</code></td><td><code>SSAOidString</code></td></tr><tr><td><code>SSAOidZero</code></td><td><code>SSAStringCpy</code></td></tr><tr><td><code>SSAStringInit</code></td><td><code>SSAStringToChar</code></td></tr><tr><td><code>SSAStringZero</code></td><td></td></tr></table>	<code>SSAOidCmp</code>	<code>SSAOidCpy</code>	<code>SSAOidDup</code>	<code>SSAOidFree</code>	<code>SSAOidInit</code>	<code>SSAOidNew</code>	<code>SSAOidStrToOid</code>	<code>SSAOidString</code>	<code>SSAOidZero</code>	<code>SSAStringCpy</code>	<code>SSAStringInit</code>	<code>SSAStringToChar</code>	<code>SSAStringZero</code>	
<code>SSAOidCmp</code>	<code>SSAOidCpy</code>														
<code>SSAOidDup</code>	<code>SSAOidFree</code>														
<code>SSAOidInit</code>	<code>SSAOidNew</code>														
<code>SSAOidStrToOid</code>	<code>SSAOidString</code>														
<code>SSAOidZero</code>	<code>SSAStringCpy</code>														
<code>SSAStringInit</code>	<code>SSAStringToChar</code>														
<code>SSAStringZero</code>															
FILES	<table><tr><td><code>/usr/lib/libssasnm.so.1</code></td><td>shared object</td></tr><tr><td><code>/usr/lib/64/libssasnm.so.1</code></td><td>64-bit shared object</td></tr></table>	<code>/usr/lib/libssasnm.so.1</code>	shared object	<code>/usr/lib/64/libssasnm.so.1</code>	64-bit shared object										
<code>/usr/lib/libssasnm.so.1</code>	shared object														
<code>/usr/lib/64/libssasnm.so.1</code>	64-bit shared object														
ATTRIBUTES	<p>See <code>attributes(5)</code> 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>Availability</td><td>SUNWsasnm</td></tr><tr><td>MT-Level</td><td>Unsafe</td></tr></tbody></table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	Availability	SUNWsasnm	MT-Level	Unsafe								
ATTRIBUTE TYPE	ATTRIBUTE VALUE														
Availability	SUNWsasnm														
MT-Level	Unsafe														
SEE ALSO	<code>intro(3)</code> , <code>libssagent(3LIB)</code> , <code>attributes(5)</code>														

NAME	libsys – system library		
SYNOPSIS	cc [<i>flag</i> . . .] <i>file</i> . . . -lsys [<i>library</i> . . .]		
DESCRIPTION	Functions in this library provide basic system services. This library is implemented as a filter on the C library (see libc(3LIB)).		
INTERFACES	The shared object <code>libsys.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.		
	<code>__ctype</code>	<code>__huge_val</code>	<code>_access</code>
	<code>_acct</code>	<code>_alarm</code>	<code>_altzone</code>
	<code>_catclose</code>	<code>_catgets</code>	<code>_catopen</code>
	<code>_chdir</code>	<code>_chmod</code>	<code>_chown</code>
	<code>_chroot</code>	<code>_close</code>	<code>_closedir</code>
	<code>_creat</code>	<code>_daylight</code>	<code>_dup</code>
	<code>_environ</code>	<code>_execl</code>	<code>_execle</code>
	<code>_execlp</code>	<code>_execv</code>	<code>_execve</code>
	<code>_execvp</code>	<code>_exit</code>	<code>_fattach</code>
	<code>_fchdir</code>	<code>_fchmod</code>	<code>_fchown</code>
	<code>_fcntl</code>	<code>_fdetach</code>	<code>_fork</code>
	<code>_fpathconf</code>	<code>_fstat</code>	<code>_fstatvfs</code>
	<code>_fsync</code>	<code>_ftok</code>	<code>_getcontext</code>
	<code>_getcwd</code>	<code>_getegid</code>	<code>_geteuid</code>
	<code>_getgid</code>	<code>_getgrgid</code>	<code>_getgrnam</code>
	<code>_getgroups</code>	<code>_getlogin</code>	<code>_getmsg</code>
	<code>_getpgid</code>	<code>_getpgrp</code>	<code>_getpid</code>
	<code>_getpmsg</code>	<code>_getppid</code>	<code>_getpwnam</code>
	<code>_getpwuid</code>	<code>_getrlimit</code>	<code>_getsid</code>
	<code>_gettxt</code>	<code>_getuid</code>	<code>_grantpt</code>
	<code>_initgroups</code>	<code>_ioctl</code>	<code>_isastream</code>
	<code>_kill</code>	<code>_lchown</code>	<code>_link</code>
	<code>_lseek</code>	<code>_lstat</code>	<code>_makecontext</code>
	<code>_memcntl</code>	<code>_mkdir</code>	<code>_mknod</code>

libsys(3LIB)

_mlock	_mmap	_mount
_mprotect	_msgctl	_msgget
_msgrcv	_msgsnd	_msync
_munlock	_munmap	_nice
_numeric	_open	_opendir
_pathconf	_pause	_pipe
_poll	_profil	_ptrace
_ptsname	_putmsg	_putpmsg
_read	_readdir	_readlink
_readv	_rename	_rewinddir
_rmdir	_seekdir	_semctl
_semget	_semop	_setcontext
_setgid	_setgroups	_setpgid
_setpgrp	_setrlimit	_setsid
_setuid	_shmat	_shmctl
_shmdt	_shmget	_sigaction
_sigaddset	_sigaltstack	_sigdelset
_sigemptyset	_sigfillset	_sighold
_sigignore	_sigismember	_siglongjmp
_sigpause	_sigpending	_sigprocmask
_sigrelse	_sigsend	_sigsendset
_sigset	_sigsetjmp	_sigsuspend
_stat	_statvfs	_stime
_swapcontext	_symlink	_sync
_sysconf	_telldir	_time
_times	_timezone	_ttyname
_tzname	_ulimit	_umask
_umount	_uname	_unlink
_unlockpt	_utime	_wait
_waitid	_waitpid	_write

_writev	access	acct
alarm	atexit	calloc
catclose	catgets	catopen
chdir	chmod	chown
chroot	close	closedir
creat	daylight	dup
environ	execl	execle
execlp	execv	execve
execvp	exit	fattach
fchdir	fchmod	fchown
fcntl	fdetach	fork
fpathconf	free	fstat
fstatvfs	fsync	ftok
getcontext	getcwd	getegid
geteuid	getgid	getgrgid
getgrnam	getgroups	getlogin
getmsg	getpgid	getpgrp
getpid	getpmsg	getppid
getpwnam	getpwuid	getrlimit
getsid	gettxt	getuid
grantpt	initgroups	ioctl
isastream	kill	lchown
link	localeconv	lseek
lstat	makecontext	malloc
memcntl	mkdir	mknod
mlock	mmap	mount
mprotect	msgctl	msgget
msgrcv	msgsnd	msync
munlock	munmap	nice
open	opendir	pathconf

libsys(3LIB)

pause	pipe	poll
profil	ptrace	ptsname
putmsg	putpmsg	read
readdir	readlink	readv
realloc	remove	rename
rewinddir	rmdir	seekdir
semctl	semget	semop
setcontext	setgid	setgroups
setlocale	setpgid	setpgrp
setrlimit	setsid	setuid
shmat	shmctl	shmdt
shmget	sigaction	sigaddset
sigaltstack	sigdelset	sigemptyset
sigfillset	sighold	sigignore
sigismember	siglongjmp	signal
sigpause	sigpending	sigprocmask
sigrelse	sigsend	sigsendset
sigset	sigsetjmp	sigsuspend
stat	statvfs	stime
strcoll	strerror	strftime
strxfrm	swapcontext	symlink
sync	sysconf	system
telldir	time	times
timezone	ttyname	tzname
ulimit	umask	umount
uname	unlink	unlockpt
utime	wait	waitid
waitpid	write	writev

The following interfaces are unique to the SPARC version of this library:

```
.div                .mul                .rem
.stret1            .stret2            .stret4
.stret8            .udiv              .umul
.urem              __Q_add            __Q_cmp
__Q_cmpe           __Q_div            __Q_dtoq
__Q_feq            __Q_fge            __Q_fgt
__Q_fle            __Qflt            __Q_fne
__Q_itoq           __Q_mul            __Q_neg
__Q_qtod           __Q_qtoi           __Q_qtos
__Q_qtou           __Q_sqrt           __Q_stoq
__Q_sub            __Q_utoq           __dtou
__ftou
```

The following interfaces are unique to the x86 version of this library:

```
__flt_rounds       __fp_hw             __fpstart
__fxstat            __lxstat            __nuname
_sbrk               __xmknod            __xstat
nuname              sbrk
```

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

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl
MT-Level	Safe

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

libsysevent(3LIB)

NAME	libsysevent – system event interface library								
SYNOPSIS	<pre>cc [<i>flag</i> . . .] <i>file</i> . . . -lsysevent [<i>library</i> . . .] #include <sysevent.h></pre>								
DESCRIPTION	Functions in this library extract specific identifier, publisher, and attribute information from a system event (sysevent) handle, defined as <code>sysevent_t</code> , and allow privileged user-level applications to queue system events for delivery to the system event daemon, <code>syseventd(1M)</code> .								
INTERFACES	The shared object <code>libsysevent.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.								
	<pre>sysevent_bind_handle sysevent_free sysevent_get_attr_list sysevent_get_class_name sysevent_get_pid sysevent_get_pub_name sysevent_get_seq sysevent_get_size sysevent_get_subclass_name sysevent_get_time sysevent_get_vendor_name sysevent_post_event sysevent_subscribe_event sysevent_unbind_handle sysevent_unsubscribe_event</pre>								
FILES	<pre>/usr/lib/libsysevent.so.1 shared object /usr/lib/64/libsysevent.so.1 64-bit shared object</pre>								
ATTRIBUTES	See <code>attributes(5)</code> for descriptions of the following attributes:								
	<table border="1"><thead><tr><th>ATTRIBUTE TYPE</th><th>ATTRIBUTE VALUE</th></tr></thead><tbody><tr><td>Availability</td><td>SUNWcsl (32-bit) SUNWcslx (64-bit)</td></tr><tr><td>Interface Stability</td><td>Evolving</td></tr><tr><td>MT-Level</td><td>MT-Safe</td></tr></tbody></table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)	Interface Stability	Evolving	MT-Level	MT-Safe
ATTRIBUTE TYPE	ATTRIBUTE VALUE								
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)								
Interface Stability	Evolving								
MT-Level	MT-Safe								
SEE ALSO	<code>syseventd(1M)</code> , <code>intro(3)</code> , <code>attributes(5)</code>								

NAME libtermcap – terminal independent operation library

SYNOPSIS `cc [flag . . .] -I /usr/ucbinclude file . . . -L /usr/libucb \`
`-R /usr/libucb -ltermcap [library . . .]`

DESCRIPTION Functions in this library extract and use capabilities from the terminal capability database terminfo(4).

INTERFACES The shared object libtermcap.so.1 provides the public interfaces defined below. See intro(3) for additional information on shared object interfaces.

BC	PC	UP	ospeed	tgetent
tgetflag	tgetnum	tgetstr	tgoto	tputs

FILES /usr/ucblib/libtermcap.a
 archive library

/usr/ucblib/libtermcap.so.1
 shared object

/usr/ucblib/64/libtermcap.so.1
 64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT-Level	Unsafe

SEE ALSO intro(3), curs_termcap(3CURSES), terminfo(4), attributes(5)

libthread(3LIB)

NAME	libthread – threads library																																																
SYNOPSIS	<code>cc -mt [<i>flag</i> . . .] <i>file</i> . . . [<i>library</i> . . .]</code>																																																
DESCRIPTION	Functions in libthread provide routines that provide threading support.																																																
INTERFACES	The shared object libthread.so.1 provides the public interfaces defined below. See intro(3) for additional information on shared object interfaces.																																																
	<table><tr><td>__pthread_cleanup_pop</td><td>__pthread_cleanup_push</td></tr><tr><td>__xpg4_putmsg</td><td>__xpg4_putpmsg</td></tr><tr><td>_getfp</td><td>_mutex_held</td></tr><tr><td>_rw_read_held</td><td>_rw_write_held</td></tr><tr><td>_sema_held</td><td>alarm</td></tr><tr><td>close</td><td>cond_broadcast</td></tr><tr><td>cond_destroy</td><td>cond_init</td></tr><tr><td>cond_reltimedwait</td><td>cond_signal</td></tr><tr><td>cond_timedwait</td><td>cond_wait</td></tr><tr><td>creat</td><td>fcntl</td></tr><tr><td>fork</td><td>fork1</td></tr><tr><td>fsync</td><td>getmsg</td></tr><tr><td>getpmsg</td><td>kill</td></tr><tr><td>lockf</td><td>lwp_self</td></tr><tr><td>msgrcv</td><td>msgsnd</td></tr><tr><td>msync</td><td>mutex_destroy</td></tr><tr><td>mutex_init</td><td>mutex_lock</td></tr><tr><td>mutex_trylock</td><td>mutex_unlock</td></tr><tr><td>open</td><td>pause</td></tr><tr><td>poll</td><td>pread</td></tr><tr><td>pthread_attr_destroy</td><td>pthread_attr_getdetachstate</td></tr><tr><td>pthread_attr_getguardsize</td><td>pthread_attr_getinheritsched</td></tr><tr><td>pthread_attr_getschedparam</td><td>pthread_attr_getschedpolicy</td></tr><tr><td>pthread_attr_getscope</td><td>pthread_attr_getstackaddr</td></tr></table>	__pthread_cleanup_pop	__pthread_cleanup_push	__xpg4_putmsg	__xpg4_putpmsg	_getfp	_mutex_held	_rw_read_held	_rw_write_held	_sema_held	alarm	close	cond_broadcast	cond_destroy	cond_init	cond_reltimedwait	cond_signal	cond_timedwait	cond_wait	creat	fcntl	fork	fork1	fsync	getmsg	getpmsg	kill	lockf	lwp_self	msgrcv	msgsnd	msync	mutex_destroy	mutex_init	mutex_lock	mutex_trylock	mutex_unlock	open	pause	poll	pread	pthread_attr_destroy	pthread_attr_getdetachstate	pthread_attr_getguardsize	pthread_attr_getinheritsched	pthread_attr_getschedparam	pthread_attr_getschedpolicy	pthread_attr_getscope	pthread_attr_getstackaddr
__pthread_cleanup_pop	__pthread_cleanup_push																																																
__xpg4_putmsg	__xpg4_putpmsg																																																
_getfp	_mutex_held																																																
_rw_read_held	_rw_write_held																																																
_sema_held	alarm																																																
close	cond_broadcast																																																
cond_destroy	cond_init																																																
cond_reltimedwait	cond_signal																																																
cond_timedwait	cond_wait																																																
creat	fcntl																																																
fork	fork1																																																
fsync	getmsg																																																
getpmsg	kill																																																
lockf	lwp_self																																																
msgrcv	msgsnd																																																
msync	mutex_destroy																																																
mutex_init	mutex_lock																																																
mutex_trylock	mutex_unlock																																																
open	pause																																																
poll	pread																																																
pthread_attr_destroy	pthread_attr_getdetachstate																																																
pthread_attr_getguardsize	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_setguardsize
pthread_attr_setinheritsched	pthread_attr_setschedparam
pthread_attr_setschedpolicy	pthread_attr_setscope
pthread_attr_setstackaddr	pthread_attr_setstacksize
pthread_cancel	pthread_cond_broadcast
pthread_cond_destroy	pthread_cond_init
pthread_cond_reltimedwait_np	pthread_cond_signal
pthread_cond_timedwait	pthread_cond_wait
pthread_condattr_destroy	pthread_condattr_getpshared
pthread_condattr_init	pthread_condattr_setpshared
pthread_create	pthread_detach
pthread_equal	pthread_exit
pthread_getconcurrency	pthread_getschedparam
pthread_getspecific	pthread_join
pthread_key_create	pthread_key_delete
pthread_kill	pthread_mutex_consistent_np
pthread_mutex_destroy	pthread_mutex_getprioceiling
pthread_mutex_init	pthread_mutex_lock
pthread_mutex_setprioceiling	pthread_mutex_trylock
pthread_mutex_unlock	pthread_mutexattr_destroy
pthread_mutexattr_getprioceiling	pthread_mutexattr_getprotocol
pthread_mutexattr_getpshared	pthread_mutexattr_getrobust_np
pthread_mutexattr_gettype	pthread_mutexattr_init
pthread_mutexattr_setprioceiling	pthread_mutexattr_setprotocol
pthread_mutexattr_setpshared	pthread_mutexattr_setrobust_np
pthread_mutexattr_settype	pthread_once
pthread_rwlock_destroy	pthread_rwlock_init
pthread_rwlock_rdlock	pthread_rwlock_tryrdlock
pthread_rwlock_trywrlock	pthread_rwlock_unlock

libthread(3LIB)

pthread_rwlock_wrlock	pthread_rwlockattr_destroy
pthread_rwlockattr_getpshared	pthread_rwlockattr_init
pthread_rwlockattr_setpshared	pthread_self
pthread_setcancelstate	pthread_setcanceltype
pthread_setconcurrency	pthread_setschedparam
pthread_setspecific	pthread_sigmask
pthread_testcancel	putmsg
putpmsg	pwrite
read	readv
rw_rdlock	rw_tryrdlock
rw_trywrlock	rw_unlock
rw_wrlock	rwlock_destroy
rwlock_init	select
sema_destroy	sema_init
sema_post	sema_trywait
sema_wait	setcontext
setitimer	sigaction
siglongjmp	sigpause
sigpending	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
usleep	wait
wait3	waitid
waitpid	write
writev	

The following interfaces are unique to the 32-bit version of this library:

creat64	lockf64
open64	pread64
pwrite64	

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

/usr/lib/64/libthread.so.1
64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)
MT-Level	Safe

SEE ALSO `pvs(1)`, `intro(2)`, `intro(3)`, `libpthread(3LIB)`, `libthread_db(3LIB)`, `libthread_db(3THR)`, `threads(3THR)`, `attributes(5)`

libthread_db(3LIB)

NAME	libthread_db – threads debugging library																																														
SYNOPSIS	<pre>cc [<i>flag</i> . . .] <i>file</i> . . . -lthread_db [<i>library</i> . . .] #include <proc_service.h> #include <thread_db.h></pre>																																														
DESCRIPTION	Functions in this library are used to build debuggers for multithreaded programs.																																														
INTERFACES	The shared object <code>libthread_db.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.																																														
	<table><tr><td><code>td_init</code></td><td><code>td_log</code></td></tr><tr><td><code>td_sync_get_info</code></td><td><code>td_sync_get_stats</code></td></tr><tr><td><code>td_sync_setstate</code></td><td><code>td_sync_waiters</code></td></tr><tr><td><code>td_ta_clear_event</code></td><td><code>td_ta_delete</code></td></tr><tr><td><code>td_ta_enable_stats</code></td><td><code>td_ta_event_addr</code></td></tr><tr><td><code>td_ta_event_getmsg</code></td><td><code>td_ta_get_nthreads</code></td></tr><tr><td><code>td_ta_get_ph</code></td><td><code>td_ta_get_stats</code></td></tr><tr><td><code>td_ta_map_addr2sync</code></td><td><code>td_ta_map_id2thr</code></td></tr><tr><td><code>td_ta_map_lwp2thr</code></td><td><code>td_ta_new</code></td></tr><tr><td><code>td_ta_reset_stats</code></td><td><code>td_ta_set_event</code></td></tr><tr><td><code>td_ta_setconcurrency</code></td><td><code>td_ta_sync_iter</code></td></tr><tr><td><code>td_ta_sync_tracking_enable</code></td><td><code>td_ta_thr_iter</code></td></tr><tr><td><code>td_ta_tsd_iter</code></td><td><code>td_thr_clear_event</code></td></tr><tr><td><code>td_thr_dbresume</code></td><td><code>td_thr_dbsuspend</code></td></tr><tr><td><code>td_thr_event_enable</code></td><td><code>td_thr_event_getmsg</code></td></tr><tr><td><code>td_thr_get_info</code></td><td><code>td_thr_getfpregs</code></td></tr><tr><td><code>td_thr_getgregs</code></td><td><code>td_thr_getxregs</code></td></tr><tr><td><code>td_thr_getxregsize</code></td><td><code>td_thr_lockowner</code></td></tr><tr><td><code>td_thr_set_event</code></td><td><code>td_thr_setfpregs</code></td></tr><tr><td><code>td_thr_setgregs</code></td><td><code>td_thr_setprio</code></td></tr><tr><td><code>td_thr_setsigpending</code></td><td><code>td_thr_setxregs</code></td></tr><tr><td><code>td_thr_sigsetmask</code></td><td><code>td_thr_sleepinfo</code></td></tr><tr><td><code>td_thr_tsd</code></td><td><code>td_thr_validate</code></td></tr></table>	<code>td_init</code>	<code>td_log</code>	<code>td_sync_get_info</code>	<code>td_sync_get_stats</code>	<code>td_sync_setstate</code>	<code>td_sync_waiters</code>	<code>td_ta_clear_event</code>	<code>td_ta_delete</code>	<code>td_ta_enable_stats</code>	<code>td_ta_event_addr</code>	<code>td_ta_event_getmsg</code>	<code>td_ta_get_nthreads</code>	<code>td_ta_get_ph</code>	<code>td_ta_get_stats</code>	<code>td_ta_map_addr2sync</code>	<code>td_ta_map_id2thr</code>	<code>td_ta_map_lwp2thr</code>	<code>td_ta_new</code>	<code>td_ta_reset_stats</code>	<code>td_ta_set_event</code>	<code>td_ta_setconcurrency</code>	<code>td_ta_sync_iter</code>	<code>td_ta_sync_tracking_enable</code>	<code>td_ta_thr_iter</code>	<code>td_ta_tsd_iter</code>	<code>td_thr_clear_event</code>	<code>td_thr_dbresume</code>	<code>td_thr_dbsuspend</code>	<code>td_thr_event_enable</code>	<code>td_thr_event_getmsg</code>	<code>td_thr_get_info</code>	<code>td_thr_getfpregs</code>	<code>td_thr_getgregs</code>	<code>td_thr_getxregs</code>	<code>td_thr_getxregsize</code>	<code>td_thr_lockowner</code>	<code>td_thr_set_event</code>	<code>td_thr_setfpregs</code>	<code>td_thr_setgregs</code>	<code>td_thr_setprio</code>	<code>td_thr_setsigpending</code>	<code>td_thr_setxregs</code>	<code>td_thr_sigsetmask</code>	<code>td_thr_sleepinfo</code>	<code>td_thr_tsd</code>	<code>td_thr_validate</code>
<code>td_init</code>	<code>td_log</code>																																														
<code>td_sync_get_info</code>	<code>td_sync_get_stats</code>																																														
<code>td_sync_setstate</code>	<code>td_sync_waiters</code>																																														
<code>td_ta_clear_event</code>	<code>td_ta_delete</code>																																														
<code>td_ta_enable_stats</code>	<code>td_ta_event_addr</code>																																														
<code>td_ta_event_getmsg</code>	<code>td_ta_get_nthreads</code>																																														
<code>td_ta_get_ph</code>	<code>td_ta_get_stats</code>																																														
<code>td_ta_map_addr2sync</code>	<code>td_ta_map_id2thr</code>																																														
<code>td_ta_map_lwp2thr</code>	<code>td_ta_new</code>																																														
<code>td_ta_reset_stats</code>	<code>td_ta_set_event</code>																																														
<code>td_ta_setconcurrency</code>	<code>td_ta_sync_iter</code>																																														
<code>td_ta_sync_tracking_enable</code>	<code>td_ta_thr_iter</code>																																														
<code>td_ta_tsd_iter</code>	<code>td_thr_clear_event</code>																																														
<code>td_thr_dbresume</code>	<code>td_thr_dbsuspend</code>																																														
<code>td_thr_event_enable</code>	<code>td_thr_event_getmsg</code>																																														
<code>td_thr_get_info</code>	<code>td_thr_getfpregs</code>																																														
<code>td_thr_getgregs</code>	<code>td_thr_getxregs</code>																																														
<code>td_thr_getxregsize</code>	<code>td_thr_lockowner</code>																																														
<code>td_thr_set_event</code>	<code>td_thr_setfpregs</code>																																														
<code>td_thr_setgregs</code>	<code>td_thr_setprio</code>																																														
<code>td_thr_setsigpending</code>	<code>td_thr_setxregs</code>																																														
<code>td_thr_sigsetmask</code>	<code>td_thr_sleepinfo</code>																																														
<code>td_thr_tsd</code>	<code>td_thr_validate</code>																																														

FILES /usr/lib/libthread_db.so.1
shared object

/usr/lib/64/libthread_db.so.1
64-bit shared object

ATTRIBUTES See attributes(5) for description of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)
MT Level	Safe

SEE ALSO pvs(1), intro(3), libpthread(3LIB), libthread(3LIB), libthread_db(3THR), threads(3THR), attributes(5)

libtnfctl(3LIB)

NAME libtnfctl – TNF probe control library

SYNOPSIS `cc [flag . . .] file . . . -ltnfctl [library . . .]
#include <tnf/tnfctl.h>`

DESCRIPTION Functions in this library provide TNF probe control routines for use by processes and the kernel.

INTERFACES The shared object `libtnfctl.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

<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>	

FILES `/usr/lib/libtnfctl.so.1`
shared object

`/usr/lib/64/libtnfctl.so.1`
64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWtnfc (32-bit) SUNWtnfcx (64-bit)

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT Level	MT-Safe with exceptions

SEE ALSO pvs(1), intro(3), libtnfctl(3TNF), tracing(3TNF), attributes(5)

NOTES This API is MT-Safe. Multiple threads can concurrently operate on independent `tnfctl` handles, which is the typical behavior expected. `libtnfctl` does not support multiple threads operating on the same `tnfctl` handle. If this is desired, it is the client's responsibility to implement locking to ensure that two threads that use the same `tnfctl` handle are not simultaneously present in a `libtnfctl` interface.

libucb(3LIBUCB)

NAME	libucb – UCB source compatibility library
SYNOPSIS	<pre>cc [<i>flag</i> . . .] -I /usr/ucbinclude <i>file</i> . . . -L /usr/libucb \ -R /usr/libucb -lucb [<i>library</i> . . .]</pre>
DESCRIPTION	Functions in this library provide UCB source compatibility.
INTERFACES	The shared object <code>libucb.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.

alphasort	bcmp	bcopy
bzero	flock	fopen
fprintf	freopen	fstatfs
ftime	getdtablesize	gethostid
gethostname	getpagesize	getrusage
gettimeofday	getwd	index
killpg	longjmp	mctl
nice	nlist	printf
psignal	rand	re_comp
re_exec	readdir	reboot
rindex	scandir	setbuffer
sethostname	setjmp	setlinebuf
setpgrp	settimeofday	sigblock
siginterrupt	signal	sigpause
sigsetmask	sigstack	sigvec
sigvechandler	sleep	sprintf
srand	statfs	sys_siglist
times	ualarm	usignal
usigpause	usleep	vfprintf
vprintf	vsprintf	wait3
wait4		

The following interfaces are unique to the 32-bit version of this library:

alphasort64 fopen64 freopen64
 readdir64 scandir64

FILES /usr/ucblib/libucb.a
 archive library
 /usr/ucblib/libucb.so.1
 shared object
 /usr/ucblib/64/libucb.so.1
 64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWscpu, SUNWsra (32-bit) SUNWscpux (64-bit)
MT-Level	Safe with exceptions

SEE ALSO pvs(1), intro(3), attributes(5)

libumem(3LIB)

NAME	libumem – object-caching memory allocation library														
SYNOPSIS	<pre>cc [<i>flag...</i>] <i>file...</i> -lumem [<i>library...</i>] #include <umem.h></pre>														
DESCRIPTION	<p>Functions in this library provide fast, scalable object-caching memory allocation with multithreaded application support. In addition to the standard <code>malloc(3C)</code> family of functions and the more flexible <code>umem_alloc(3MALLOC)</code> family, <code>libumem</code> provides powerful object-caching services as described in <code>umem_cache_create(3MALLOC)</code>.</p> <p>The <code>libumem</code> library also provides extensive debugging support, including detection of memory leaks, buffer overruns, multiple frees, use of uninitialized data, use of freed data, and many other common programming errors. See <code>umem_debug(3MALLOC)</code>.</p>														
INTERFACES	<p>The shared object <code>libumem.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.</p> <table><tr><td><code>calloc</code></td><td><code>free</code></td></tr><tr><td><code>malloc</code></td><td><code>memalign</code></td></tr><tr><td><code>realloc</code></td><td><code>umem_alloc</code></td></tr><tr><td><code>umem_cache_alloc</code></td><td><code>umem_cache_create</code></td></tr><tr><td><code>umem_cache_destroy</code></td><td><code>umem_cache_free</code></td></tr><tr><td><code>umem_free</code></td><td><code>umem_nofail_callback</code></td></tr><tr><td><code>umem_zalloc</code></td><td><code>valloc</code></td></tr></table>	<code>calloc</code>	<code>free</code>	<code>malloc</code>	<code>memalign</code>	<code>realloc</code>	<code>umem_alloc</code>	<code>umem_cache_alloc</code>	<code>umem_cache_create</code>	<code>umem_cache_destroy</code>	<code>umem_cache_free</code>	<code>umem_free</code>	<code>umem_nofail_callback</code>	<code>umem_zalloc</code>	<code>valloc</code>
<code>calloc</code>	<code>free</code>														
<code>malloc</code>	<code>memalign</code>														
<code>realloc</code>	<code>umem_alloc</code>														
<code>umem_cache_alloc</code>	<code>umem_cache_create</code>														
<code>umem_cache_destroy</code>	<code>umem_cache_free</code>														
<code>umem_free</code>	<code>umem_nofail_callback</code>														
<code>umem_zalloc</code>	<code>valloc</code>														
FILES	<table><tr><td><code>/usr/lib/libumem.so.1</code></td><td>shared object</td></tr><tr><td><code>/usr/lib/64/libumem.so.1</code></td><td>64-bit shared object</td></tr></table>	<code>/usr/lib/libumem.so.1</code>	shared object	<code>/usr/lib/64/libumem.so.1</code>	64-bit shared object										
<code>/usr/lib/libumem.so.1</code>	shared object														
<code>/usr/lib/64/libumem.so.1</code>	64-bit shared object														
ATTRIBUTES	See <code>attributes(5)</code> for descriptions of the following attributes:														
	<table border="1"><thead><tr><th>ATTRIBUTE TYPE</th><th>ATTRIBUTE VALUE</th></tr></thead><tbody><tr><td>Availability</td><td>SUNWcsl (32-bit) SUNWcslx (64-bit)</td></tr><tr><td>Interface Stability</td><td>Evolving</td></tr><tr><td>MT-Level</td><td>MT-Safe</td></tr></tbody></table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)	Interface Stability	Evolving	MT-Level	MT-Safe						
ATTRIBUTE TYPE	ATTRIBUTE VALUE														
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)														
Interface Stability	Evolving														
MT-Level	MT-Safe														
SEE ALSO	<code>intro(3)</code> , <code>malloc(3C)</code> , <code>umem_alloc(3MALLOC)</code> , <code>umem_cache_create(3MALLOC)</code> , <code>umem_debug(3MALLOC)</code> , <code>attributes(5)</code>														

NAME libvolmgt – volume management library

SYNOPSIS `cc [flag . . .] file . . . -lvolmgt [library . . .]`
`#include <volmgt.h>`

DESCRIPTION Functions in this library provide access to the volume management services.

INTERFACES The shared object `libvolmgt.so.1` provides the public interfaces defined below. See `intro(3)` for additional information on shared object interfaces.

<code>media_findname</code>	<code>media_getattr</code>
<code>media_getid</code>	<code>media_setattr</code>
<code>volmgt_acquire</code>	<code>volmgt_check</code>
<code>volmgt_feature_enabled</code>	<code>volmgt_inuse</code>
<code>volmgt_ownspath</code>	<code>volmgt_release</code>
<code>volmgt_root</code>	<code>volmgt_running</code>
<code>volmgt_symdev</code>	<code>volmgt_symname</code>

FILES `/usr/lib/libvolmgt.a`
 archive library

`/usr/lib/libvolmgt.so.1`
 shared object

`/usr/lib/64/libvolmgt.so.1`
 64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl, SUNWarc (32-bit) SUNWcslx (64-bit)
MT-Level	Safe with exceptions

SEE ALSO `pvs(1)`, `intro(3)`, `media_findname(3VOLMGT)`, `attributes(5)`

NOTES The MT-Level for this library of interfaces is Safe, except for `media_findname(3VOLMGT)`, which is Unsafe.

libw(3LIB)

NAME	libw – wide character library																																																												
SYNOPSIS	<pre>cc [flag . . .] file . . . [library . . .] #include <wchar.h></pre>																																																												
DESCRIPTION	<p>Historically, functions in this library provided wide character translations. This functionality now resides in libc(3LIB).</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 libw.so.1, and the archive version is implemented as a null archive. New application development need not reference either version of libw.</p>																																																												
INTERFACES	<p>The shared object libw.so.1 provides the public interfaces defined below. See intro(3) for additional information on shared object interfaces.</p> <table><tr><td>fgetwc</td><td>fgetws</td><td>fputwc</td></tr><tr><td>fputws</td><td>getwc</td><td>getwchar</td></tr><tr><td>getws</td><td>isenglish</td><td>isideogram</td></tr><tr><td>isnumber</td><td>isphonogram</td><td>isspecial</td></tr><tr><td>iswalnum</td><td>iswalpha</td><td>iswcntrl</td></tr><tr><td>iswctype</td><td>iswdigit</td><td>iswgraph</td></tr><tr><td>iswlower</td><td>iswprint</td><td>iswpunct</td></tr><tr><td>iswspace</td><td>iswupper</td><td>iswxdigit</td></tr><tr><td>putwc</td><td>putwchar</td><td>putws</td></tr><tr><td>strtows</td><td>towlower</td><td>towupper</td></tr><tr><td>ungetwc</td><td>watoll</td><td>wscat</td></tr><tr><td>wcschr</td><td>wscmp</td><td>wscoll</td></tr><tr><td>wcscpy</td><td>wcscspn</td><td>wcsftime</td></tr><tr><td>wcslen</td><td>wcsncat</td><td>wcsncmp</td></tr><tr><td>wcsncpy</td><td>wcspbrk</td><td>wcsrchr</td></tr><tr><td>wcsspn</td><td>wcstod</td><td>wcstok</td></tr><tr><td>wcstol</td><td>wcstoul</td><td>wcswcs</td></tr><tr><td>wcswidth</td><td>wcsxfrm</td><td>wctype</td></tr><tr><td>wcwidth</td><td>wscasecmp</td><td>wscat</td></tr><tr><td>wchr</td><td>wscmp</td><td>wscol</td></tr></table>	fgetwc	fgetws	fputwc	fputws	getwc	getwchar	getws	isenglish	isideogram	isnumber	isphonogram	isspecial	iswalnum	iswalpha	iswcntrl	iswctype	iswdigit	iswgraph	iswlower	iswprint	iswpunct	iswspace	iswupper	iswxdigit	putwc	putwchar	putws	strtows	towlower	towupper	ungetwc	watoll	wscat	wcschr	wscmp	wscoll	wcscpy	wcscspn	wcsftime	wcslen	wcsncat	wcsncmp	wcsncpy	wcspbrk	wcsrchr	wcsspn	wcstod	wcstok	wcstol	wcstoul	wcswcs	wcswidth	wcsxfrm	wctype	wcwidth	wscasecmp	wscat	wchr	wscmp	wscol
fgetwc	fgetws	fputwc																																																											
fputws	getwc	getwchar																																																											
getws	isenglish	isideogram																																																											
isnumber	isphonogram	isspecial																																																											
iswalnum	iswalpha	iswcntrl																																																											
iswctype	iswdigit	iswgraph																																																											
iswlower	iswprint	iswpunct																																																											
iswspace	iswupper	iswxdigit																																																											
putwc	putwchar	putws																																																											
strtows	towlower	towupper																																																											
ungetwc	watoll	wscat																																																											
wcschr	wscmp	wscoll																																																											
wcscpy	wcscspn	wcsftime																																																											
wcslen	wcsncat	wcsncmp																																																											
wcsncpy	wcspbrk	wcsrchr																																																											
wcsspn	wcstod	wcstok																																																											
wcstol	wcstoul	wcswcs																																																											
wcswidth	wcsxfrm	wctype																																																											
wcwidth	wscasecmp	wscat																																																											
wchr	wscmp	wscol																																																											

wscoll	wscopy	wscspn
wsdup	wslen	wncasecmp
wscat	wncmp	wncpy
wspbrk	wsprintf	wsrchr
wscanf	wspn	wstod
wstok	wstol	wstoll
wstostr	wxfrm	

FILES /usr/lib/libw.a a link to /usr/lib/null.a
 /usr/lib/libw.so.1 a filter on libc.so.1
 /usr/lib/64/libw.so.1 a filter on 64/libc.so.1

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl, SUNWarc (32-bit) SUNWcslx (64-bit)
MT-Level	Safe

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

libwsreg(3LIB)

NAME	libwsreg – product install registry library																																																
SYNOPSIS	<pre>cc [<i>flag</i> . . .] <i>file</i> . . . -lwsreg [<i>library</i> . . .] #include <wsreg.h></pre>																																																
DESCRIPTION	Functions in this library provide access to the product install registry.																																																
INTERFACES	The shared object <code>libwsreg.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.																																																
	<table><tr><td><code>wsreg_add_child_component</code></td><td><code>wsreg_add_compatible_version</code></td></tr><tr><td><code>wsreg_add_dependent_component</code></td><td><code>wsreg_add_display_name</code></td></tr><tr><td><code>wsreg_add_required_component</code></td><td><code>wsreg_can_access_registry</code></td></tr><tr><td><code>wsreg_clone_component</code></td><td><code>wsreg_components_equal</code></td></tr><tr><td><code>wsreg_create_component</code></td><td><code>wsreg_free_component</code></td></tr><tr><td><code>wsreg_free_component_array</code></td><td><code>wsreg_get</code></td></tr><tr><td><code>wsreg_get_all</code></td><td><code>wsreg_get_child_components</code></td></tr><tr><td><code>wsreg_get_compatible_versions</code></td><td><code>wsreg_get_data</code></td></tr><tr><td><code>wsreg_get_data_pairs</code></td><td><code>wsreg_get_dependent_components</code></td></tr><tr><td><code>wsreg_get_display_languages</code></td><td><code>wsreg_get_display_name</code></td></tr><tr><td><code>wsreg_get_id</code></td><td><code>wsreg_get_instance</code></td></tr><tr><td><code>wsreg_get_location</code></td><td><code>wsreg_get_parent</code></td></tr><tr><td><code>wsreg_get_required_components</code></td><td><code>wsreg_get_type</code></td></tr><tr><td><code>wsreg_get_uninstaller</code></td><td><code>wsreg_get_unique_name</code></td></tr><tr><td><code>wsreg_get_vendor</code></td><td><code>wsreg_get_version</code></td></tr><tr><td><code>wsreg_initialize</code></td><td><code>wsreg_query_create</code></td></tr><tr><td><code>wsreg_query_free</code></td><td><code>wsreg_query_get_id</code></td></tr><tr><td><code>wsreg_query_get_instance</code></td><td><code>wsreg_query_get_location</code></td></tr><tr><td><code>wsreg_query_get_unique_name</code></td><td><code>wsreg_query_get_version</code></td></tr><tr><td><code>wsreg_query_set_id</code></td><td><code>wsreg_query_set_instance</code></td></tr><tr><td><code>wsreg_query_set_location</code></td><td><code>wsreg_query_set_unique_name</code></td></tr><tr><td><code>wsreg_query_set_version</code></td><td><code>wsreg_register</code></td></tr><tr><td><code>wsreg_remove_child_component</code></td><td><code>wsreg_remove_compatible_version</code></td></tr><tr><td><code>wsreg_remove_dependent_component</code></td><td><code>wsreg_remove_display_name</code></td></tr></table>	<code>wsreg_add_child_component</code>	<code>wsreg_add_compatible_version</code>	<code>wsreg_add_dependent_component</code>	<code>wsreg_add_display_name</code>	<code>wsreg_add_required_component</code>	<code>wsreg_can_access_registry</code>	<code>wsreg_clone_component</code>	<code>wsreg_components_equal</code>	<code>wsreg_create_component</code>	<code>wsreg_free_component</code>	<code>wsreg_free_component_array</code>	<code>wsreg_get</code>	<code>wsreg_get_all</code>	<code>wsreg_get_child_components</code>	<code>wsreg_get_compatible_versions</code>	<code>wsreg_get_data</code>	<code>wsreg_get_data_pairs</code>	<code>wsreg_get_dependent_components</code>	<code>wsreg_get_display_languages</code>	<code>wsreg_get_display_name</code>	<code>wsreg_get_id</code>	<code>wsreg_get_instance</code>	<code>wsreg_get_location</code>	<code>wsreg_get_parent</code>	<code>wsreg_get_required_components</code>	<code>wsreg_get_type</code>	<code>wsreg_get_uninstaller</code>	<code>wsreg_get_unique_name</code>	<code>wsreg_get_vendor</code>	<code>wsreg_get_version</code>	<code>wsreg_initialize</code>	<code>wsreg_query_create</code>	<code>wsreg_query_free</code>	<code>wsreg_query_get_id</code>	<code>wsreg_query_get_instance</code>	<code>wsreg_query_get_location</code>	<code>wsreg_query_get_unique_name</code>	<code>wsreg_query_get_version</code>	<code>wsreg_query_set_id</code>	<code>wsreg_query_set_instance</code>	<code>wsreg_query_set_location</code>	<code>wsreg_query_set_unique_name</code>	<code>wsreg_query_set_version</code>	<code>wsreg_register</code>	<code>wsreg_remove_child_component</code>	<code>wsreg_remove_compatible_version</code>	<code>wsreg_remove_dependent_component</code>	<code>wsreg_remove_display_name</code>
<code>wsreg_add_child_component</code>	<code>wsreg_add_compatible_version</code>																																																
<code>wsreg_add_dependent_component</code>	<code>wsreg_add_display_name</code>																																																
<code>wsreg_add_required_component</code>	<code>wsreg_can_access_registry</code>																																																
<code>wsreg_clone_component</code>	<code>wsreg_components_equal</code>																																																
<code>wsreg_create_component</code>	<code>wsreg_free_component</code>																																																
<code>wsreg_free_component_array</code>	<code>wsreg_get</code>																																																
<code>wsreg_get_all</code>	<code>wsreg_get_child_components</code>																																																
<code>wsreg_get_compatible_versions</code>	<code>wsreg_get_data</code>																																																
<code>wsreg_get_data_pairs</code>	<code>wsreg_get_dependent_components</code>																																																
<code>wsreg_get_display_languages</code>	<code>wsreg_get_display_name</code>																																																
<code>wsreg_get_id</code>	<code>wsreg_get_instance</code>																																																
<code>wsreg_get_location</code>	<code>wsreg_get_parent</code>																																																
<code>wsreg_get_required_components</code>	<code>wsreg_get_type</code>																																																
<code>wsreg_get_uninstaller</code>	<code>wsreg_get_unique_name</code>																																																
<code>wsreg_get_vendor</code>	<code>wsreg_get_version</code>																																																
<code>wsreg_initialize</code>	<code>wsreg_query_create</code>																																																
<code>wsreg_query_free</code>	<code>wsreg_query_get_id</code>																																																
<code>wsreg_query_get_instance</code>	<code>wsreg_query_get_location</code>																																																
<code>wsreg_query_get_unique_name</code>	<code>wsreg_query_get_version</code>																																																
<code>wsreg_query_set_id</code>	<code>wsreg_query_set_instance</code>																																																
<code>wsreg_query_set_location</code>	<code>wsreg_query_set_unique_name</code>																																																
<code>wsreg_query_set_version</code>	<code>wsreg_register</code>																																																
<code>wsreg_remove_child_component</code>	<code>wsreg_remove_compatible_version</code>																																																
<code>wsreg_remove_dependent_component</code>	<code>wsreg_remove_display_name</code>																																																

wsreg_remove_required_component	wsreg_set_data
wsreg_set_id	wsreg_set_instance
wsreg_set_location	wsreg_set_parent
wsreg_set_type	wsreg_set_uninstaller
wsreg_set_unique_name	wsreg_set_vendor
wsreg_set_version	wsreg_unregister

FILES /usr/lib/libwsreg.so.1
shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWwsr2
MT-Level	Unsafe

SEE ALSO prodreg(1M), intro(3), attributes(5)

libxfn(3LIB)

NAME	libxfn – X/Open Federated Naming (XFN) library
SYNOPSIS	<pre>cc [<i>flag</i> . . .] <i>file</i> . . . -lxfn [<i>library</i> . . .] #include <xfn/xfn.h></pre>
DESCRIPTION	Functions in this library provide the implementation of XFN, the X/Open Federated Naming specification (see xfn(3XFN) and fns(5)).
INTERFACES	The shared object <code>libxfn.so.2</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.

<code>_pure_error_</code>	<code>fn_attr_bind</code>
<code>fn_attr_create_subcontext</code>	<code>fn_attr_get</code>
<code>fn_attr_get_ids</code>	<code>fn_attr_get_values</code>
<code>fn_attr_modify</code>	<code>fn_attr_multi_get</code>
<code>fn_attr_multi_modify</code>	<code>fn_attribute_add</code>
<code>fn_attribute_assign</code>	<code>fn_attribute_copy</code>
<code>fn_attribute_create</code>	<code>fn_attribute_destroy</code>
<code>fn_attribute_first</code>	<code>fn_attribute_identifier</code>
<code>fn_attribute_next</code>	<code>fn_attribute_remove</code>
<code>fn_attribute_syntax</code>	<code>fn_attribute_valuecount</code>
<code>fn_attrmodlist_add</code>	<code>fn_attrmodlist_assign</code>
<code>fn_attrmodlist_copy</code>	<code>fn_attrmodlist_count</code>
<code>fn_attrmodlist_create</code>	<code>fn_attrmodlist_destroy</code>
<code>fn_attrmodlist_first</code>	<code>fn_attrmodlist_next</code>
<code>fn_attrset_add</code>	<code>fn_attrset_assign</code>
<code>fn_attrset_copy</code>	<code>fn_attrset_count</code>
<code>fn_attrset_create</code>	<code>fn_attrset_destroy</code>
<code>fn_attrset_first</code>	<code>fn_attrset_get</code>
<code>fn_attrset_next</code>	<code>fn_attrset_remove</code>
<code>fn_bindinglist_destroy</code>	<code>fn_bindinglist_next</code>
<code>fn_composite_name_append_comp</code>	<code>fn_composite_name_append_name</code>
<code>fn_composite_name_assign</code>	<code>fn_composite_name_copy</code>
<code>fn_composite_name_count</code>	<code>fn_composite_name_create</code>

fn_composite_name_delete_comp	fn_composite_name_destroy
fn_composite_name_first	fn_composite_name_from_str
fn_composite_name_from_string	fn_composite_name_insert_comp
fn_composite_name_insert_name	fn_composite_name_is_empty
fn_composite_name_is_equal	fn_composite_name_is_prefix
fn_composite_name_is_suffix	fn_composite_name_last
fn_composite_name_next	fn_composite_name_prefix
fn_composite_name_prepend_comp	fn_composite_name_prepend_name
fn_composite_name_prev	fn_composite_name_suffix
fn_compound_name_append_comp	fn_compound_name_assign
fn_compound_name_copy	fn_compound_name_count
fn_compound_name_delete_all	fn_compound_name_delete_comp
fn_compound_name_destroy	fn_compound_name_first
fn_compound_name_from_syntax_attrs	fn_compound_name_get_syntax_attrs
fn_compound_name_insert_comp	fn_compound_name_is_empty
fn_compound_name_is_equal	fn_compound_name_is_prefix
fn_compound_name_is_suffix	fn_compound_name_last
fn_compound_name_next	fn_compound_name_prefix
fn_compound_name_prepend_comp	fn_compound_name_prev
fn_compound_name_suffix	fn_ctx_bind
fn_ctx_create_subcontext	fn_ctx_destroy_subcontext
fn_ctx_get_ref	fn_ctx_get_syntax_attrs
fn_ctx_handle_destroy	fn_ctx_handle_from_initial
fn_ctx_handle_from_ref	fn_ctx_list_bindings
fn_ctx_list_names	fn_ctx_lookup
fn_ctx_lookup_link	fn_ctx_rename
fn_ctx_unbind	fn_multigetlist_destroy
fn_multigetlist_next	fn_namelist_destroy
fn_namelist_next	fn_ref_addr_assign
fn_ref_addr_copy	fn_ref_addr_create

libxfn(3LIB)

fn_ref_addr_data	fn_ref_addr_description
fn_ref_addr_destroy	fn_ref_addr_length
fn_ref_addr_type	fn_ref_addrcount
fn_ref_append_addr	fn_ref_assign
fn_ref_copy	fn_ref_create
fn_ref_create_link	fn_ref_delete_addr
fn_ref_delete_all	fn_ref_description
fn_ref_destroy	fn_ref_first
fn_ref_insert_addr	fn_ref_is_link
fn_ref_link_name	fn_ref_next
fn_ref_prepend_addr	fn_ref_type
fn_status_advance_by_name	fn_status_append_remaining_name
fn_status_append_resolved_name	fn_status_assign
fn_status_code	fn_status_copy
fn_status_create	fn_status_description
fn_status_destroy	fn_status_diagnostic_message
fn_status_is_success	fn_status_link_code
fn_status_link_diagnostic_message	fn_status_link_remaining_name
fn_status_link_resolved_name	fn_status_link_resolved_ref
fn_status_remaining_name	fn_status_resolved_name
fn_status_resolved_ref	fn_status_set
fn_status_set_code	fn_status_set_diagnostic_message
fn_status_set_link_code	fn_status_set_link_diagnostic_message
fn_status_set_link_remaining_name	fn_status_set_link_resolved_name
fn_status_set_link_resolved_ref	fn_status_set_remaining_name
fn_status_set_resolved_name	fn_status_set_resolved_ref
fn_status_set_success	fn_string_assign
fn_string_bytecount	fn_string_charcount
fn_string_code_set	fn_string_compare
fn_string_compare_substring	fn_string_contents

fn_string_copy	fn_string_create
fn_string_destroy	fn_string_from_composite_name
fn_string_from_compound_name	fn_string_from_contents
fn_string_from_str	fn_string_from_str_n
fn_string_from_strings	fn_string_from_substring
fn_string_is_empty	fn_string_lang_err
fn_string_next_substring	fn_string_prev_substring
fn_string_str	fn_valuelist_destroy
fn_valuelist_next	prelim_fn_attr_ext_search
prelim_fn_attr_search	prelim_fn_ctx_equivalent_name
prelim_fn_ext_searchlist_destroy	prelim_fn_ext_searchlist_next
prelim_fn_search_control_assign	prelim_fn_search_control_copy
prelim_fn_search_control_create	prelim_fn_search_control_destroy
prelim_fn_search_control_follow_links	prelim_fn_search_control_max_names
prelim_fn_search_control_return_attr_ids	prelim_fn_search_control_return_ref
prelim_fn_search_control_scope	prelim_fn_search_filter_arguments
prelim_fn_search_filter_assign	prelim_fn_search_filter_copy
prelim_fn_search_filter_create	prelim_fn_search_filter_destroy
prelim_fn_search_filter_expression	prelim_fn_searchlist_destroy
prelim_fn_searchlist_next	

FILES	/usr/lib/libxfn.so.2	shared object
	/usr/lib/64/libxfn.so.2	64-bit shared object
	/usr/lib/libxfn.so.1	shared object for backward compatibility only
	/usr/lib/64/libxfn.so.1	64-bit shared object for backward compatibility only

libxfn(3LIB)

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWfns (32-bit) SUNWfnsx (64-bit)
MT-Level	Safe

SEE ALSO `pvs(1)`, `intro(3)`, `xfn(3XFN)`, `attributes(5)`, `fns(5)`

NOTES The Federated Naming Service based on the X/Open XFN standard might not be supported in a future release of the Solaris operating system.

NAME	libxnet – X/Open Networking library	
SYNOPSIS	cc [<i>flag</i> . . .] <i>file</i> . . . -lxnet [<i>library</i> . . .]	
DESCRIPTION	Functions in this library provide networking interfaces which comply with the X/Open CAE Specification, Networking Services, Issue 4.	
INTERFACES	The shared object <code>libxnet.so.1</code> provides the public interfaces defined below. See <code>intro(3)</code> for additional information on shared object interfaces.	
	<code>__t_errno</code>	<code>__xnet_bind</code>
	<code>__xnet_connect</code>	<code>__xnet_getsockopt</code>
	<code>__xnet_listen</code>	<code>__xnet_recvmsg</code>
	<code>__xnet_sendmsg</code>	<code>__xnet_sendto</code>
	<code>__xnet_socket</code>	<code>__xnet_socketpair</code>
	<code>_xti_accept</code>	<code>_xti_alloc</code>
	<code>_xti_bind</code>	<code>_xti_close</code>
	<code>_xti_connect</code>	<code>_xti_error</code>
	<code>_xti_free</code>	<code>_xti_getinfo</code>
	<code>_xti_getprotaddr</code>	<code>_xti_getstate</code>
	<code>_xti_listen</code>	<code>_xti_lock</code>
	<code>_xti_open</code>	<code>_xti_optmgmt</code>
	<code>_xti_rcv</code>	<code>_xti_rcvconnect</code>
	<code>_xti_rcvdis</code>	<code>_xti_rcvrel</code>
	<code>_xti_rcvreldata</code>	<code>_xti_rcvudata</code>
	<code>_xti_rcvuderr</code>	<code>_xti_rcvv</code>
	<code>_xti_rcvvudata</code>	<code>_xti_snd</code>
	<code>_xti_snddis</code>	<code>_xti_sndrel</code>
	<code>_xti_sndreldata</code>	<code>_xti_sndudata</code>
	<code>_xti_sndv</code>	<code>_xti_sndvudata</code>
	<code>_xti_strerror</code>	<code>_xti_sync</code>
	<code>_xti_sysconf</code>	<code>_xti_unbind</code>
	<code>_xti_xns5_accept</code>	<code>_xti_xns5_snd</code>
	<code>accept</code>	<code>bind</code>

libxnet(3LIB)

connect	endhostent
endnetent	endprotoent
endservent	gethostbyaddr
gethostbyname	gethostent
gethostname	getnetbyaddr
getnetbyname	getnetent
getpeername	getprotobyname
getprotobynumber	getprotoent
getservbyname	getservbyport
getservent	getsockname
getsockopt	h_errno
htonl	htons
inet_addr	inet_lnaof
inet_makeaddr	inet_netof
inet_network	inet_ntoa
listen	ntohl
ntohs	recv
recvfrom	recvmsg
send	sendmsg
sendto	sethostent
setnetent	setprotoent
setservent	setsockopt
shutdown	socket
socketpair	t_errno

FILES /usr/lib/libxnet.so.1 shared object
/usr/lib/64/libxnet.so.1 64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl (32-bit) SUNWcslx (64-bit)
MT-Level	Safe

SEE ALSO [intro\(3\)](#), [attributes\(5\)](#), [standards\(5\)](#)

liby(3LIB)

NAME liby – yacc library

SYNOPSIS cc [*flag* . . .] *file* . . . -ly [*library* . . .]

DESCRIPTION The function in this library provides a user interface to the yacc(1) library.

INTERFACES The shared object liby.so.1 provides the public interface defined below. See intro(3) for additional information on shared object interfaces.

yyerror

FILES /usr/lib/liby.a archive library
/usr/lib/liby.so.1 shared object
/usr/lib/sparcv9/liby.so.1 64-bit shared object

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsl, SUNWbtool (32-bit) SUNWcslx (64-bit)
MT-Level	Unsafe

SEE ALSO yacc(1), intro(3), attributes(5)

NAME	math – math functions and constants
SYNOPSIS	<code>#include <math.h></code>
DESCRIPTION	<p>This file contains declarations of all the functions in the Math Library (described in Section 3M), as well as various functions in the C Library (Section 3C) that return floating-point values.</p> <p>It defines the structure and constants used by the <code>matherr(3M)</code> error-handling mechanisms, including the following constant used as a error-return value:</p> <p><code>HUGE</code> The maximum value of a single-precision floating-point number.</p> <p>The following mathematical constants are defined for user convenience:</p> <p><code>M_E</code> The base of natural logarithms (e).</p> <p><code>M_LOG2E</code> The base-2 logarithm of e.</p> <p><code>M_LOG10E</code> The base-10 logarithm of e.</p> <p><code>M_LN2</code> The natural logarithm of 2.</p> <p><code>M_LN10</code> The natural logarithm of 10.</p> <p><code>M_PI</code> π, the ratio of the circumference of a circle to its diameter.</p> <p><code>M_PI_2</code> $\pi/2$.</p> <p><code>M_PI_4</code> $\pi/4$.</p> <p><code>M_1_PI</code> $1/\pi$.</p> <p><code>M_2_PI</code> $2/\pi$.</p> <p><code>M_2_SQRTPI</code> 2 over the square root of π.</p> <p><code>M_SQRT2</code> The positive square root of 2.</p> <p><code>M_SQRT1_2</code> The positive square root of $1/2$.</p> <p>The following mathematical constants are also defined in this header file:</p> <p><code>MAXFLOAT</code> The maximum value of a non-infinite single-precision floating point number.</p> <p><code>HUGE_VAL</code> positive infinity. For the definitions of various machine-dependent constants see <code>values(3HEAD)</code>.</p>
SEE ALSO	<code>intro(3)</code> , <code>matherr(3M)</code> , <code>values(3HEAD)</code>

mqueue(3HEAD)

NAME	mqueue – message queues												
SYNOPSIS	<pre>#include <mqueue.h></pre>												
DESCRIPTION	<p>The <code><mqueue.h></code> header defines the <code>mqd_t</code> type, which is used for message queue descriptors. This will not be an array type. A message queue descriptor may be implemented using a file descriptor, in which case applications can open up to at least <code>OPEN_MAX</code> file and message queues.</p> <p>The <code><mqueue.h></code> header defines the <code>sigevent</code> structure (as described in <code><signal.h></code>, see <code>signal(3HEAD)</code>) and the <code>mq_attr</code> structure, which is used in getting and setting the attributes of a message queue. Attributes are initially set when the message queue is created. A <code>mq_attr</code> structure has the following members:</p> <table><tr><td>long</td><td><code>mq_flags</code></td><td>message queue flags</td></tr><tr><td>long</td><td><code>mq_maxmsg</code></td><td>maximum number of messages</td></tr><tr><td>long</td><td><code>mq_msgsize</code></td><td>maximum message size</td></tr><tr><td>long</td><td><code>mq_curmsgs</code></td><td>number of messages currently queued</td></tr></table> <p>Inclusion of the <code><mqueue.h></code> header may make visible symbols defined in the headers <code><fcntl.h></code>, <code><signal.h></code>, <code><sys/types.h></code>, and <code><time.h></code>.</p>	long	<code>mq_flags</code>	message queue flags	long	<code>mq_maxmsg</code>	maximum number of messages	long	<code>mq_msgsize</code>	maximum message size	long	<code>mq_curmsgs</code>	number of messages currently queued
long	<code>mq_flags</code>	message queue flags											
long	<code>mq_maxmsg</code>	maximum number of messages											
long	<code>mq_msgsize</code>	maximum message size											
long	<code>mq_curmsgs</code>	number of messages currently queued											
SEE ALSO	<code>fcntl(3HEAD)</code> , <code>signal(3HEAD)</code> , <code>time(3HEAD)</code> , <code>types(3HEAD)</code>												

NAME	ndbm – definitions for ndbm database operations
SYNOPSIS	<pre>#include <ndbm.h></pre>
DESCRIPTION	<p>The <code><ndbm.h></code> header defines the <code>datum</code> type as a structure that includes at least the following members:</p> <pre>void *dptr /* pointer to the application's data */ size_t dsize /* size of the object pointed to by dptr */</pre> <p>The <code>size_t</code> type is defined through <code>typedef</code> as described in <code><stddef.h></code>.</p> <p>The <code><ndbm.h></code> header defines the <code>DBM</code> type through <code>typedef</code>.</p> <p>The following constants are defined as possible values for the <code>store_mode</code> argument to <code>dbm_store()</code>:</p> <pre>DBM_INSERT Insertion of new entries only. DBM_REPLACE Allow replacing existing entries.</pre>
SEE ALSO	ndbm(3C), standards(5)

netdb(3HEAD)

NAME	netdb – definitions for network database operations																																				
SYNOPSIS	<pre>#include <netdb.h></pre>																																				
DESCRIPTION	<p>The <code><netdb.h></code> header defines the type <code>in_port_t</code> and the type <code>in_addr_t</code> as described in <code>in(3HEAD)</code>.</p> <p>The <code><netdb.h></code> header defines the <code>hostent</code> structure that includes the following members:</p> <table><tr><td><code>char</code></td><td><code>*h_name</code></td><td>Official name of the host.</td></tr><tr><td><code>char</code></td><td><code>**h_aliases</code></td><td>A pointer to an array of pointers to alternative host names, terminated by a null pointer.</td></tr><tr><td><code>int</code></td><td><code>h_addrtype</code></td><td>Address type.</td></tr><tr><td><code>int</code></td><td><code>h_length</code></td><td>The length, in bytes, of the address.</td></tr><tr><td><code>char</code></td><td><code>**h_addr_list</code></td><td>A pointer to an array of pointers to network addresses (in network byte order) for the host, terminated by a null pointer.</td></tr></table> <p>The <code><netdb.h></code> header defines the <code>netent</code> structure that includes the following members:</p> <table><tr><td><code>char</code></td><td><code>*n_name</code></td><td>Official, fully-qualified (including the domain) name of the network.</td></tr><tr><td><code>char</code></td><td><code>**n_aliases</code></td><td>A pointer to an array of pointers to alternative network names, terminated by a null pointer.</td></tr><tr><td><code>int</code></td><td><code>n_addrtype</code></td><td>The address type of the network.</td></tr><tr><td><code>in_addr_t</code></td><td><code>n_net</code></td><td>The network number, in host byte order.</td></tr></table> <p>The <code><netdb.h></code> header defines the <code>protoent</code> structure that includes the following members:</p> <table><tr><td><code>char</code></td><td><code>*p_name</code></td><td>Official name of the protocol.</td></tr><tr><td><code>char</code></td><td><code>**p_aliases</code></td><td>A pointer to an array of pointers to alternative protocol names, terminated by a null pointer.</td></tr><tr><td><code>int</code></td><td><code>p_proto</code></td><td>The protocol number.</td></tr></table> <p>The <code><netdb.h></code> header defines the <code>servent</code> structure that includes the following members:</p>	<code>char</code>	<code>*h_name</code>	Official name of the host.	<code>char</code>	<code>**h_aliases</code>	A pointer to an array of pointers to alternative host names, terminated by a null pointer.	<code>int</code>	<code>h_addrtype</code>	Address type.	<code>int</code>	<code>h_length</code>	The length, in bytes, of the address.	<code>char</code>	<code>**h_addr_list</code>	A pointer to an array of pointers to network addresses (in network byte order) for the host, terminated by a null pointer.	<code>char</code>	<code>*n_name</code>	Official, fully-qualified (including the domain) name of the network.	<code>char</code>	<code>**n_aliases</code>	A pointer to an array of pointers to alternative network names, terminated by a null pointer.	<code>int</code>	<code>n_addrtype</code>	The address type of the network.	<code>in_addr_t</code>	<code>n_net</code>	The network number, in host byte order.	<code>char</code>	<code>*p_name</code>	Official name of the protocol.	<code>char</code>	<code>**p_aliases</code>	A pointer to an array of pointers to alternative protocol names, terminated by a null pointer.	<code>int</code>	<code>p_proto</code>	The protocol number.
<code>char</code>	<code>*h_name</code>	Official name of the host.																																			
<code>char</code>	<code>**h_aliases</code>	A pointer to an array of pointers to alternative host names, terminated by a null pointer.																																			
<code>int</code>	<code>h_addrtype</code>	Address type.																																			
<code>int</code>	<code>h_length</code>	The length, in bytes, of the address.																																			
<code>char</code>	<code>**h_addr_list</code>	A pointer to an array of pointers to network addresses (in network byte order) for the host, terminated by a null pointer.																																			
<code>char</code>	<code>*n_name</code>	Official, fully-qualified (including the domain) name of the network.																																			
<code>char</code>	<code>**n_aliases</code>	A pointer to an array of pointers to alternative network names, terminated by a null pointer.																																			
<code>int</code>	<code>n_addrtype</code>	The address type of the network.																																			
<code>in_addr_t</code>	<code>n_net</code>	The network number, in host byte order.																																			
<code>char</code>	<code>*p_name</code>	Official name of the protocol.																																			
<code>char</code>	<code>**p_aliases</code>	A pointer to an array of pointers to alternative protocol names, terminated by a null pointer.																																			
<code>int</code>	<code>p_proto</code>	The protocol number.																																			

char	*s_name	Official name of the service.
char	**s_aliases	A pointer to an array of pointers to alternative service names, terminated by a null pointer.
int	s_port	The port number at which the service resides, in network byte order.
char	*s_proto	The name of the protocol to use when contacting the service.

The <netdb.h> header defines the macro `IPPORT_RESERVED` with the value of the highest reserved Internet port number.

The <netdb.h> header provides a declaration for `h_errno`:

```
extern int h_errno;
```

The <netdb.h> header defines the following macros for use as error values for `gethostbyaddr()` and `gethostbyname()`:

```
HOST_NOT_FOUND      NO_DATA
NO_RECOVERY         TRY_AGAIN
```

Inclusion of the <netdb.h> header may also make visible all symbols from `in(3HEAD)`.

Default For applications that do not require standard-conforming behavior (those that use the socket interfaces described in section 3N of the reference manual; see `Intro(3)` and `standards(5)`), the following are declared as functions and can also be defined as macros:

```
int          endhostent(void);
int          endnetent(void);
int          endprotoent(void);
int          endservent(void);
struct hostent *gethostbyaddr(const void *addr, int len, int type);
struct hostent *gethostbyname(const char *name);
struct hostent *gethostent(void);
struct netent *getnetbyaddr(long net, int type);
```

netdb(3HEAD)

```
struct netent      *getnetbyname(const char *name);
struct netent      *getnetent(void);
struct protoent    *getprotobyname(const char *name);
struct protoent    *getprotobynumber(int proto);
struct protoent    *getprotoent(void);
struct servent     *getservbyname(const char *name, const char *proto);
struct servent     *getservbyport(int port, const char *proto);
struct servent     *getservent(void);
int                sethostent(int stayopen);
int                setnetent(int stayopen);
int                setprotoent(int stayopen);
int                setservent(int stayopen);
```

Standard conforming

For applications that require standard-conforming behavior (those that use the socket interfaces described in section 3XN of the reference manual; see [Intro\(3\)](#) and [standards\(5\)](#)), the following are declared as functions and can also be defined as macros:

```
void               endhostent(void);
void               endnetent(void);
void               endprotoent(void);
void               endservent(void);
struct hostent     *gethostbyaddr(const void *addr, size_t len, int type);
struct hostent     *gethostbyname(const char *name);
struct hostent     *gethostent(void);
struct netent      *getnetbyaddr(in_addr_t net, int type);
struct netent      *getnetbyname(const char *name);
struct netent      *getnetent(void);
struct protoent    *getprotobyname(const char *name);
struct protoent    *getprotobynumber(int proto);
struct protoent    *getprotoent(void);
struct servent     *getservbyname(const char *name, const char *proto);
```

netdb(3HEAD)

```
struct servent      *getservbyport (int port, const char *proto);
struct servent      *getservent (void) ;
void                sethostent (int stayopen);
void                setnetent (int stayopen);
void                setprotoent (int stayopen);
void                setservent (int stayopen);
```

SEE ALSO Intro(3), endhostent(3NSL), endhostent(3XNET), endnetent(3SOCKET), endnetent(3XNET), endprotoent(3SOCKET), endprotoent(3XNET), endservent(3SOCKET), endservent(3XNET), in(3HEAD), standards(5)

nl_types(3HEAD)

NAME	nl_types – native language data types												
SYNOPSIS	#include <nl_types.h>												
DESCRIPTION	<p>This header contains the following definitions:</p> <table><tr><td>nl_catd</td><td>Used by the message catalog functions <code>catopen</code>, <code>catgets</code> and <code>catclose</code> to identify a catalog.</td></tr><tr><td>nl_item</td><td>Used by <code>nl_langinfo</code> to identify items of <code>langinfo</code> data. Values for objects of type <code>nl_item</code> are defined in <code><langinfo.h></code>.</td></tr><tr><td>NL_SETD</td><td>Used by <code>gencat</code> when no <code>\$set</code> directive is specified in a message text source file. This constant can be used in subsequent calls to <code>catgets</code> as the value of the set identifier parameter.</td></tr><tr><td>NL_MSGSMAX</td><td>Maximum number of messages per set.</td></tr><tr><td>NL_SETMAX</td><td>Maximum number of sets per catalog.</td></tr><tr><td>NL_TEXTMAX</td><td>Maximum size of a message.</td></tr></table>	nl_catd	Used by the message catalog functions <code>catopen</code> , <code>catgets</code> and <code>catclose</code> to identify a catalog.	nl_item	Used by <code>nl_langinfo</code> to identify items of <code>langinfo</code> data. Values for objects of type <code>nl_item</code> are defined in <code><langinfo.h></code> .	NL_SETD	Used by <code>gencat</code> when no <code>\$set</code> directive is specified in a message text source file. This constant can be used in subsequent calls to <code>catgets</code> as the value of the set identifier parameter.	NL_MSGSMAX	Maximum number of messages per set.	NL_SETMAX	Maximum number of sets per catalog.	NL_TEXTMAX	Maximum size of a message.
nl_catd	Used by the message catalog functions <code>catopen</code> , <code>catgets</code> and <code>catclose</code> to identify a catalog.												
nl_item	Used by <code>nl_langinfo</code> to identify items of <code>langinfo</code> data. Values for objects of type <code>nl_item</code> are defined in <code><langinfo.h></code> .												
NL_SETD	Used by <code>gencat</code> when no <code>\$set</code> directive is specified in a message text source file. This constant can be used in subsequent calls to <code>catgets</code> as the value of the set identifier parameter.												
NL_MSGSMAX	Maximum number of messages per set.												
NL_SETMAX	Maximum number of sets per catalog.												
NL_TEXTMAX	Maximum size of a message.												
SEE ALSO	<code>gencat(1)</code> , <code>catgets(3C)</code> , <code>catopen(3C)</code> , <code>nl_langinfo(3C)</code> , <code>langinfo(3HEAD)</code>												

NAME	sched – execution scheduling								
SYNOPSIS	#include <sched.h>								
DESCRIPTION	<p>The <sched.h> header defines the <code>sched_param</code> structure, which contains the scheduling parameters required for implementation of each supported scheduling policy. This structure contains the following member:</p> <pre>int sched_priority process execution scheduling priority</pre> <p>Each process is controlled by an associated scheduling policy and priority. Associated with each policy is a priority range. Each policy definition specifies the minimum priority range for that policy. The priority ranges for each policy may overlap the priority ranges of other policies.</p> <p>The scheduling policies are indicated by the values of the following symbolic constants:</p> <table border="0"> <tr> <td style="vertical-align: top;">SCHED_FIFO</td> <td>Processes are scheduled according to the First-In-First-Out (FIFO) policy. Processes scheduled to this policy, if not pre-empted by a higher priority or interrupted by a signal, will proceed until completion.</td> </tr> <tr> <td style="vertical-align: top;">SCHED_RR</td> <td>Processes are scheduled according to the Round-Robin (RR) policy. Processes scheduled to this policy, if not pre-empted by a higher priority or interrupted by a signal, will execute for a time period, returned by <code>sched_rr_get_interval(3RT)</code> or by the system.</td> </tr> <tr> <td style="vertical-align: top;">SCHED_IA</td> <td>Processes are scheduled according to the Inter-Active Class (IA) policy as described in <code>prctl(2)</code>.</td> </tr> <tr> <td style="vertical-align: top;">SCHED_OTHER</td> <td>Processes are scheduled according to another policy not described above.</td> </tr> </table> <p>The values of these constants are distinct.</p> <p>Inclusion of the <sched.h> header will make visible symbols defined in the header <time.h>.</p>	SCHED_FIFO	Processes are scheduled according to the First-In-First-Out (FIFO) policy. Processes scheduled to this policy, if not pre-empted by a higher priority or interrupted by a signal, will proceed until completion.	SCHED_RR	Processes are scheduled according to the Round-Robin (RR) policy. Processes scheduled to this policy, if not pre-empted by a higher priority or interrupted by a signal, will execute for a time period, returned by <code>sched_rr_get_interval(3RT)</code> or by the system.	SCHED_IA	Processes are scheduled according to the Inter-Active Class (IA) policy as described in <code>prctl(2)</code> .	SCHED_OTHER	Processes are scheduled according to another policy not described above.
SCHED_FIFO	Processes are scheduled according to the First-In-First-Out (FIFO) policy. Processes scheduled to this policy, if not pre-empted by a higher priority or interrupted by a signal, will proceed until completion.								
SCHED_RR	Processes are scheduled according to the Round-Robin (RR) policy. Processes scheduled to this policy, if not pre-empted by a higher priority or interrupted by a signal, will execute for a time period, returned by <code>sched_rr_get_interval(3RT)</code> or by the system.								
SCHED_IA	Processes are scheduled according to the Inter-Active Class (IA) policy as described in <code>prctl(2)</code> .								
SCHED_OTHER	Processes are scheduled according to another policy not described above.								
SEE ALSO	<code>prctl(2)</code> , <code>sched_rr_get_interval(3RT)</code> , <code>time(3HEAD)</code>								

siginfo(3HEAD)

NAME	siginfo – signal generation information												
SYNOPSIS	<pre>#include <siginfo.h></pre>												
DESCRIPTION	<p>If a process is catching a signal, it might request information that tells why the system generated that signal. See <code>sigaction(2)</code>. If a process is monitoring its children, it might receive information that tells why a child changed state. See <code>waitid(2)</code>. In either case, the system returns the information in a structure of type <code>siginfo_t</code>, which includes the following information:</p> <pre>int si_signo /* signal number */ int si_errno /* error number */ int si_code /* signal code */ union signal si_value /* signal value */</pre> <p><code>si_signo</code> contains the system-generated signal number. For the <code>waitid(2)</code> function, <code>si_signo</code> is always <code>SIGCHLD</code>.</p> <p>If <code>si_errno</code> is non-zero, it contains an error number associated with this signal, as defined in <code><errno.h></code>.</p> <p><code>si_code</code> contains a code identifying the cause of the signal.</p> <p>If the value of the <code>si_code</code> member is <code>SI_NOINFO</code>, only the <code>si_signo</code> member of <code>siginfo_t</code> is meaningful, and the value of all other members is unspecified.</p> <p>User Signals If the value of <code>si_code</code> is less than or equal to 0, then the signal was generated by a user process (see <code>kill(2)</code>, <code>_lwp_kill(2)</code>, <code>sigqueue(3RT)</code>, <code>sigsend(2)</code>, <code>abort(3C)</code>, and <code>raise(3C)</code>) and the <code>siginfo</code> structure contains the following additional information:</p> <pre>typedef long pid_t si_pid /* sending process ID */ typedef long uid_t si_uid /* sending user ID */</pre> <p>If the signal was generated by a user process, the following values are defined for <code>si_code</code>:</p> <table><tr><td><code>SI_USER</code></td><td>The implementation sets <code>si_code</code> to <code>SI_USER</code> if the signal was sent by <code>kill(2)</code>, <code>sigsend(2)</code>, <code>raise(3C)</code> or <code>abort(3C)</code>.</td></tr><tr><td><code>SI_LWP</code></td><td>The signal was sent by <code>_lwp_kill(2)</code>.</td></tr><tr><td><code>SI_QUEUE</code></td><td>The signal was sent by <code>sigqueue(3RT)</code>.</td></tr><tr><td><code>SI_TIMER</code></td><td>The signal was generated by the expiration of a timer created by <code>timer_settime(3RT)</code>.</td></tr><tr><td><code>SI_ASYNCIO</code></td><td>The signal was generated by the completion of an asynchronous I/O request.</td></tr><tr><td><code>SI_MSGQ</code></td><td>The signal was generated by the arrival of a message on an empty message queue. See <code>mq_notify(3RT)</code>.</td></tr></table>	<code>SI_USER</code>	The implementation sets <code>si_code</code> to <code>SI_USER</code> if the signal was sent by <code>kill(2)</code> , <code>sigsend(2)</code> , <code>raise(3C)</code> or <code>abort(3C)</code> .	<code>SI_LWP</code>	The signal was sent by <code>_lwp_kill(2)</code> .	<code>SI_QUEUE</code>	The signal was sent by <code>sigqueue(3RT)</code> .	<code>SI_TIMER</code>	The signal was generated by the expiration of a timer created by <code>timer_settime(3RT)</code> .	<code>SI_ASYNCIO</code>	The signal was generated by the completion of an asynchronous I/O request.	<code>SI_MSGQ</code>	The signal was generated by the arrival of a message on an empty message queue. See <code>mq_notify(3RT)</code> .
<code>SI_USER</code>	The implementation sets <code>si_code</code> to <code>SI_USER</code> if the signal was sent by <code>kill(2)</code> , <code>sigsend(2)</code> , <code>raise(3C)</code> or <code>abort(3C)</code> .												
<code>SI_LWP</code>	The signal was sent by <code>_lwp_kill(2)</code> .												
<code>SI_QUEUE</code>	The signal was sent by <code>sigqueue(3RT)</code> .												
<code>SI_TIMER</code>	The signal was generated by the expiration of a timer created by <code>timer_settime(3RT)</code> .												
<code>SI_ASYNCIO</code>	The signal was generated by the completion of an asynchronous I/O request.												
<code>SI_MSGQ</code>	The signal was generated by the arrival of a message on an empty message queue. See <code>mq_notify(3RT)</code> .												

System Signals

`si_value` contains the application specified value, which is passed to the application's signal-catching function at the time of the signal delivery if `si_code` is any of `SI_QUEUE`, `SI_TIMER`, `SI_ASYNCIO`, or `SI_MESGQ`.

Non-user generated signals can arise for a number of reasons. For all of these cases, `si_code` contains a positive value reflecting the reason why the system generated the signal:

Signal	Code	Reason
SIGILL	ILL_ILLOPC	illegal opcode
	ILL_ILLOPN	illegal operand
	ILL_ILLADR	illegal addressing mode
	ILL_ILLTRP	illegal trap
	ILL_PRVOPC	privileged opcode
	ILL_PRVREG	privileged register
	ILL_COPROC	co-processor error
	ILL_BADSTK	internal stack error
SIGFPE	FPE_INTDIV	integer divide by zero
	FPE_INTOVF	integer overflow
	FPE_FLTDIV	floating point divide by zero
	FPE_FLTOVF	floating point overflow
	FPE_FLTUND	floating point underflow
	FPE_FLTRES	floating point inexact result
	FPE_FLTINV	invalid floating point operation
	FPE_FLTSUB	subscript out of range
SIGSEGV	SEGV_MAPERR	address not mapped to object
	SEGV_ACCERR	invalid permissions for mapped object
SIGBUS	BUS_ADRALN	invalid address alignment
	BUS_ADRERR	non-existent physical address
	BUS_OBJERR	object specific hardware error
SIGTRAP	TRAP_BRKPT	process breakpoint
	TRAP_TRACE	process trace trap
SIGCHLD	CLD_EXITED	child has exited

siginfo(3HEAD)

	CLD_KILLED	child was killed
	CLD_DUMPED	child terminated abnormally
	CLD_TRAPPED	traced child has trapped
	CLD_STOPPED	child has stopped
	CLD_CONTINUED	stopped child had continued
SIGPOLL	POLL_IN	data input available
	POLL_OUT	output buffers available
	POLL_MSG	input message available
	POLL_ERR	I/O error
	POLL_PRI	high priority input available
	POLL_HUP	device disconnected

Signals can also be generated from the resource control subsystem. Where these signals do not already possess kernel-level `siginfo` codes, the `siginfo si_code` will be filled with `SI_RCTL` to indicate a kernel-generated signal from an established resource control value.

Signal	Code	Reason
SIGXRES	SI_RCTL	resource-control generated signal
SIGHUP		
SIGTERM		

The uncatchable signals `SIGSTOP` and `SIGKILL` have undefined `siginfo` codes.

Signals sent with a `siginfo` code of `SI_RCTL` contain code-dependent information for kernel-generated signals:

Code	Field	Value
SI_RCTL	hr_time si_entity	process-model entity of control

In addition, the following signal-dependent information is available for kernel-generated signals:

Signal	Field	Value
--------	-------	-------

SIGILL	caddr_t si_addr	address of faulting instruction
SIGFPE		
SIGSEGV	caddr_t si_addr	address of faulting memory reference
SIGBUS		
SIGCHLD	pid_t si_pid	child process ID
	int si_status	exit value or signal
SIGPOLL	long si_band	band event for POLL_IN, POLL_OUT, or POLL_MSG

SEE ALSO `_lwp_kill(2)`, `kill(2)`, `setrctl(2)`, `sigaction(2)`, `sigsend(2)`, `waitid(2)`, `abort(3C)`, `aio_read(3RT)`, `mq_notify(3RT)`, `raise(3C)`, `signal(3HEAD)`, `sigqueue(3RT)`, `timer_create(3RT)`, `timer_settime(3RT)`

NOTES For SIGCHLD signals, if `si_code` is equal to `CLD_EXITED` then `si_status` is equal to the exit value of the process; otherwise, it is equal to the signal that caused the process to change state. For some implementations, the exact value of `si_addr` might not be available; in that case, `si_addr` is guaranteed to be on the same page as the faulting instruction or memory reference.

signal(3HEAD)

NAME	signal – base signals
SYNOPSIS	<pre>#include <signal.h></pre>
DESCRIPTION	<p>A signal is an asynchronous notification of an event. A signal is said to be generated for (or sent to) a process when the event associated with that signal first occurs. Examples of such events include hardware faults, timer expiration and terminal activity, as well as the invocation of the <code>kill(2)</code> or <code>sigsend(2)</code> functions. In some circumstances, the same event generates signals for multiple processes. A process may request a detailed notification of the source of the signal and the reason why it was generated. See <code>siginfo(3HEAD)</code>.</p> <p>Signals can be generated synchronously or asynchronously. Events directly caused by the execution of code by a thread, such as a reference to an unmapped, protected, or bad memory can generate <code>SIGSEGV</code> or <code>SIGBUS</code>; a floating point exception can generate <code>SIGFPE</code>; and the execution of an illegal instruction can generate <code>SIGILL</code>. Such events are referred to as traps; signals generated by traps are said to be synchronously generated. Synchronously generated signals are initiated by a specific thread and are delivered to and handled by that thread.</p> <p>Signals may also be generated by calling <code>kill()</code>, <code>sigqueue()</code>, or <code>sigsend()</code>. Events such as keyboard interrupts generate signals, such as <code>SIGINT</code>, which are sent to the target process. Such events are referred to as interrupts; signals generated by interrupts are said to be asynchronously generated. Asynchronously generated signals are not directed to a particular thread but are handled by an arbitrary thread that meets either of the following conditions:</p> <ul style="list-style-type: none">■ The thread is blocked in a call to <code>sigwait(2)</code> whose argument includes the type of signal generated.■ The thread has a signal mask that does not include the type of signal generated. A process responds to signals in similar ways whether it is using threads or it is using lightweight processes (LWPs). See <code>thr_create(3THR)</code>. Each process may specify a system action to be taken in response to each signal sent to it, called the signal's disposition. All threads or LWPs in the process share the disposition. The set of system signal actions for a process is initialized from that of its parent. Once an action is installed for a specific signal, it usually remains installed until another disposition is explicitly requested by a call to either <code>sigaction()</code>, <code>signal()</code> or <code>sigset()</code>, or until the process <code>execs()</code>. See <code>sigaction(2)</code> and <code>signal(3C)</code>. When a process <code>execs</code>, all signals whose disposition has been set to catch the signal will be set to <code>SIG_DFL</code>. Alternatively, a process may request that the system automatically reset the disposition of a signal to <code>SIG_DFL</code> after it has been caught. See <code>sigaction(2)</code> and <code>signal(3C)</code>.
SIGNAL DELIVERY	<p>A signal is said to be delivered to a process when a thread or LWP within the process takes the appropriate action for the disposition of the signal. Delivery of a signal can be blocked. There are two methods for handling delivery of a signal in a multithreaded application. The first method specifies a signal handler function to execute when the signal is received by the process. See <code>sigaction(2)</code>. The second method creates a thread to handle the receipt of the signal <code>sigaction()</code> can be used</p>

for both synchronously and asynchronously generated signals. `sigwait()` will only work for asynchronously generated signals, as synchronously generated signals are sent to the thread that caused the event. `sigwait()` is the recommended interface for use with a multithreaded application. See `sigwait(2)`.

SIGNAL MASK

Each thread or LWP has a signal mask that defines the set of signals currently blocked from delivery to it. The signal mask of the main thread or LWP is inherited from the signal mask of the thread or LWP that created it in the parent process. The selection of the thread or LWP within the process that is to take the appropriate action for the signal is based on the method of signal generation and the signal masks of the threads or LWPs in the receiving process. Signals that are generated by action of a particular thread or LWP such as hardware faults are delivered to the thread or LWP that caused the signal. See `thr_sigsetmask(3THR)` or `sigprocmask(2)`. See `alarm(2)` for current semantics of delivery of `SIGALRM`. Signals that are directed to a particular thread or LWP are delivered to the targeted thread or LWP. See `thr_kill(3THR)` or `_lwp_kill(2)`. If the selected thread or LWP has blocked the signal, it remains pending on the thread or LWP until it is unblocked. For all other types of signal generation (for example, `kill(2)`, `sigsend(2)`, terminal activity, and other external events not ascribable to a particular thread or LWP) one of the threads or LWPs that does not have the signal blocked is selected to process the signal. If all the threads or LWPs within the process block the signal, it remains pending on the process until a thread or LWP in the process unblocks it. If the action associated with a signal is set to ignore the signal then both currently pending and subsequently generated signals of this type are discarded immediately for this process.

The determination of which action is taken in response to a signal is made at the time the signal is delivered to a thread or LWP within the process, allowing for any changes since the time of generation. This determination is independent of the means by which the signal was originally generated.

The signals currently defined by `<signal.h>` are as follows:

Name	Value	Default	Event
SIGHUP	1	Exit	Hangup (see <code>termio(7I)</code>)
SIGINT	2	Exit	Interrupt (see <code>termio(7I)</code>)
SIGQUIT	3	Core	Quit (see <code>termio(7I)</code>)
SIGILL	4	Core	Illegal Instruction
SIGTRAP	5	Core	Trace or Breakpoint Trap
SIGABRT	6	Core	Abort
SIGEMT	7	Core	Emulation Trap
SIGFPE	8	Core	Arithmetic Exception

signal(3HEAD)

Name	Value	Default	Event
SIGKILL	9	Exit	Killed
SIGBUS	10	Core	Bus Error
SIGSEGV	11	Core	Segmentation Fault
SIGSYS	12	Core	Bad System Call
SIGPIPE	13	Exit	Broken Pipe
SIGALRM	14	Exit	Alarm Clock
SIGTERM	15	Exit	Terminated
SIGUSR1	16	Exit	User Signal 1
SIGUSR2	17	Exit	User Signal 2
SIGCHLD	18	Ignore	Child Status Changed
SIGPWR	19	Ignore	Power Fail or Restart
SIGWINCH	20	Ignore	Window Size Change
SIGURG	21	Ignore	Urgent Socket Condition
SIGPOLL	22	Exit	Pollable Event (see <code>streamio(7I)</code>)
SIGSTOP	23	Stop	Stopped (signal)
SIGTSTP	24	Stop	Stopped (user) (see <code>termio(7I)</code>)
SIGCONT	25	Ignore	Continued
SIGTTIN	26	Stop	Stopped (tty input) (see <code>termio(7I)</code>)
SIGTTOU	27	Stop	Stopped (tty output) (see <code>termio(7I)</code>)
SIGVTALRM	28	Exit	Virtual Timer Expired
SIGPROF	29	Exit	Profiling Timer Expired
SIGXCPU	30	Core	CPU time limit exceeded (see <code>getrlimit(2)</code>)
SIGXFSZ	31	Core	File size limit exceeded (see <code>getrlimit(2)</code>)
SIGWAITING	32	Ignore	Concurrency signal reserved by threads library
SIGLWP	33	Ignore	Inter-LWP signal reserved by threads library
SIGFREEZE	34	Ignore	Check point Freeze
SIGTHAW	35	Ignore	Check point Thaw
SIGCANCEL	36	Ignore	Cancellation signal reserved by threads library

Name	Value	Default	Event
SIGXRES	37	Ignore	Resource control exceeded (see <code>setrctl(2)</code>)
SIGRTMIN	*	Exit	First real time signal
(SIGRTMIN+1)	*	Exit	Second real time signal
. . .			
(SIGRTMAX-1)	*	Exit	Second-to-last real time signal
SIGRTMAX	*	Exit	Last real time signal

The symbols SIGRTMIN through SIGRTMAX are evaluated dynamically in order to permit future configurability.

SIGNAL DISPOSITION

A process, using a `signal(3C)`, `sigset(3C)` or `sigaction(2)` system call, may specify one of three dispositions for a signal: take the default action for the signal, ignore the signal, or catch the signal.

Default Action: SIG_DFL

A disposition of `SIG_DFL` specifies the default action. The default action for each signal is listed in the table above and is selected from the following:

- Exit When it gets the signal, the receiving process is to be terminated with all the consequences outlined in `exit(2)`.
- Core When it gets the signal, the receiving process is to be terminated with all the consequences outlined in `exit(2)`. In addition, a “core image” of the process is constructed in the current working directory.
- Stop When it gets the signal, the receiving process is to stop. When a process is stopped, all the threads and LWPs within the process also stop executing.
- Ignore When it gets the signal, the receiving process is to ignore it. This is identical to setting the disposition to `SIG_IGN`.

Ignore Signal: SIG_IGN

A disposition of `SIG_IGN` specifies that the signal is to be ignored. Setting a signal action to `SIG_IGN` for a signal that is pending causes the pending signal to be discarded, whether or not it is blocked. Any queued values pending are also discarded, and the resources used to queue them are released and made available to queue other signals.

Catch Signal: function address

A disposition that is a function address specifies that, when it gets the signal, the thread or LWP within the process that is selected to process the signal will execute the signal handler at the specified address. Normally, the signal handler is passed the signal number as its only argument; if the disposition was set with the `sigaction()` however, additional arguments may be requested (see `sigaction(2)`). When the signal handler returns, the receiving process resumes execution at the point it was interrupted, unless the signal handler makes other arrangements. If an invalid function address is specified, results are undefined.

signal(3HEAD)

If the disposition has been set with the `sigset()` or `sigaction()`, the signal is automatically blocked in the thread or LWP while it is executing the signal catcher. If a `longjmp()` is used to leave the signal catcher, then the signal must be explicitly unblocked by the user. See `setjmp(3C)`, `signal(3C)` and `sigprocmask(2)`.

If execution of the signal handler interrupts a blocked function call, the handler is executed and the interrupted function call returns `-1` to the calling process with `errno` set to `EINTR`. However, if the `SA_RESTART` flag is set, the function call will be transparently restarted.

Some signal-generating functions, such as high resolution timer expiration, asynchronous I/O completion, inter-process message arrival, and the `sigqueue(3RT)` function, support the specification of an application defined value, either explicitly as a parameter to the function, or in a `sigevent` structure parameter. The `sigevent` structure is defined by `<signal.h>` and contains at least the following members:

Member	Member	
Type	Name	Description
int	<code>sigev_notify</code>	Notification type
int	<code>sigev_signo</code>	Signal number
union <code>sigval</code>	<code>sigev_value</code>	Signal value

The `sigval` union is defined by `<signal.h>` and contains at least the following members:

Member	Member	
Type	Name	Description
int	<code>sival_int</code>	Integer signal value
void *	<code>sival_ptr</code>	Pointer signal value

The `sigev_notify` member specifies the notification mechanism to use when an asynchronous event occurs. The `sigev_notify` member may be defined with the following values:

<code>SIGEV_NONE</code>	No asynchronous notification is delivered when the event of interest occurs.
<code>SIGEV_SIGNAL</code>	A queued signal, with its value application-defined, is generated when the event of interest occurs.

Your implementation may define additional notification mechanisms.

The `sigev_signo` member specifies the signal to be generated.

The `sigev_value` member references the application defined value to be passed to the signal-catching function at the time of the signal delivery as the `si_value` member of the `siginfo_t` structure.

The `sival_int` member is used when the application defined value is of type `int`, and the `sival_ptr` member is used when the application defined value is a pointer.

When a signal is generated by `sigqueue(3RT)` or any signal-generating function which supports the specification of an application defined value, the signal is marked pending and, if the `SA_SIGINFO` flag is set for that signal, the signal is queued to the process along with the application specified signal value. Multiple occurrences of signals so generated are queued in FIFO order. If the `SA_SIGINFO` flag is not set for that signal, later occurrences of that signal's generation, when a signal is already queued, are silently discarded.

SEE ALSO `intro(2)`, `_lwp_kill(2)`, `alarm(2)`, `exit(2)`, `getrlimit(2)`, `ioctl(2)`, `kill(2)`, `pause(2)`, `setrctl(2)`, `sigaction(2)`, `sigaltstack(2)`, `sigprocmask(2)`, `sigsend(2)`, `sigsuspend(2)`, `sigwait(2)`, `wait(2)`, `setjmp(3C)`, `siginfo(3HEAD)`, `signal(3C)`, `sigqueue(3RT)`, `sigsetops(3C)`, `thr_create(3THR)`, `thr_kill(3THR)`, `thr_sigsetmask(3THR)`, `ucontext(3HEAD)`

NOTES The dispositions of the `SIGKILL` and `SIGSTOP` signals cannot be altered from their default values. The system generates an error if this is attempted.

The `SIGKILL` and `SIGSTOP` signals cannot be blocked. The system silently enforces this restriction.

Whenever a process receives a `SIGSTOP`, `SIGTSTP`, `SIGTTIN`, or `SIGTTOU` signal, regardless of its disposition, any pending `SIGCONT` signal are discarded.

Whenever a process receives a `SIGCONT` signal, regardless of its disposition, any pending `SIGSTOP`, `SIGTSTP`, `SIGTTIN`, and `SIGTTOU` signals is discarded. In addition, if the process was stopped, it is continued.

`SIGPOLL` is issued when a file descriptor corresponding to a STREAMS file has a "selectable" event pending. See `intro(2)`. A process must specifically request that this signal be sent using the `I_SETSIG` `ioctl` call. Otherwise, the process will never receive `SIGPOLL`.

If the disposition of the `SIGCHLD` signal has been set with `signal` or `sigset`, or with `sigaction` and the `SA_NOCLDSTOP` flag has been specified, it will only be sent to the calling process when its children exit; otherwise, it will also be sent when the calling process's children are stopped or continued due to job control.

The name `SIGCLD` is also defined in this header and identifies the same signal as `SIGCHLD`. `SIGCLD` is provided for backward compatibility, new applications should use `SIGCHLD`.

signal(3HEAD)

The disposition of signals that are inherited as `SIG_IGN` should not be changed.

Signals which are generated synchronously should not be masked. If such a signal is blocked and delivered, the receiving process is killed.

NAME	socket – Internet Protocol family																																			
SYNOPSIS	#include <sys/socket.h>																																			
DESCRIPTION	<p>The <sys/socket.h> header defines the unsigned integral type <code>sa_family_t</code> through typedef.</p> <p>The <sys/socket.h> header defines the <code>sockaddr</code> structure that includes the following members:</p> <hr/> <table> <tr> <td><code>sa_family_t</code></td> <td><code>sa_family</code></td> <td><code>/* address family */</code></td> </tr> <tr> <td><code>char</code></td> <td><code>sa_data[]</code></td> <td><code>/* socket address (variable-length data) */</code></td> </tr> </table> <hr/> <p>The <sys/socket.h> header defines the <code>msghdr</code> structure that includes the following members:</p> <hr/> <table> <tr> <td><code>void</code></td> <td><code>*msg_name</code></td> <td><code>/* optional address */</code></td> </tr> <tr> <td><code>size_t</code></td> <td><code>msg_namelen</code></td> <td><code>/* size of address */</code></td> </tr> <tr> <td><code>struct iovec</code></td> <td><code>*msg_iov</code></td> <td><code>/* scatter/gather array */</code></td> </tr> <tr> <td><code>int</code></td> <td><code>msg_iovlen</code></td> <td><code>/* members in msg_iov */</code></td> </tr> <tr> <td><code>void</code></td> <td><code>*msg_control</code></td> <td><code>/* ancillary data, see below */</code></td> </tr> <tr> <td><code>size_t</code></td> <td><code>msg_controllen</code></td> <td><code>/* ancillary data buffer len */</code></td> </tr> <tr> <td><code>int</code></td> <td><code>msg_flags</code></td> <td><code>/* flags on received message */</code></td> </tr> </table> <hr/> <p>The <sys/socket.h> header defines the <code>cmsghdr</code> structure that includes the following members:</p> <hr/> <table> <tr> <td><code>size_t</code></td> <td><code>cmsg_len</code></td> <td><code>/* data byte count, including hdr */</code></td> </tr> <tr> <td><code>int</code></td> <td><code>cmsg_level</code></td> <td><code>/* originating protocol */</code></td> </tr> </table> <hr/>			<code>sa_family_t</code>	<code>sa_family</code>	<code>/* address family */</code>	<code>char</code>	<code>sa_data[]</code>	<code>/* socket address (variable-length data) */</code>	<code>void</code>	<code>*msg_name</code>	<code>/* optional address */</code>	<code>size_t</code>	<code>msg_namelen</code>	<code>/* size of address */</code>	<code>struct iovec</code>	<code>*msg_iov</code>	<code>/* scatter/gather array */</code>	<code>int</code>	<code>msg_iovlen</code>	<code>/* members in msg_iov */</code>	<code>void</code>	<code>*msg_control</code>	<code>/* ancillary data, see below */</code>	<code>size_t</code>	<code>msg_controllen</code>	<code>/* ancillary data buffer len */</code>	<code>int</code>	<code>msg_flags</code>	<code>/* flags on received message */</code>	<code>size_t</code>	<code>cmsg_len</code>	<code>/* data byte count, including hdr */</code>	<code>int</code>	<code>cmsg_level</code>	<code>/* originating protocol */</code>
<code>sa_family_t</code>	<code>sa_family</code>	<code>/* address family */</code>																																		
<code>char</code>	<code>sa_data[]</code>	<code>/* socket address (variable-length data) */</code>																																		
<code>void</code>	<code>*msg_name</code>	<code>/* optional address */</code>																																		
<code>size_t</code>	<code>msg_namelen</code>	<code>/* size of address */</code>																																		
<code>struct iovec</code>	<code>*msg_iov</code>	<code>/* scatter/gather array */</code>																																		
<code>int</code>	<code>msg_iovlen</code>	<code>/* members in msg_iov */</code>																																		
<code>void</code>	<code>*msg_control</code>	<code>/* ancillary data, see below */</code>																																		
<code>size_t</code>	<code>msg_controllen</code>	<code>/* ancillary data buffer len */</code>																																		
<code>int</code>	<code>msg_flags</code>	<code>/* flags on received message */</code>																																		
<code>size_t</code>	<code>cmsg_len</code>	<code>/* data byte count, including hdr */</code>																																		
<code>int</code>	<code>cmsg_level</code>	<code>/* originating protocol */</code>																																		

socket(3HEAD)

int	msg_type	/* protocol-specific type */
-----	----------	---------------------------------

Ancillary data consists of a sequence of pairs, each consisting of a `cmsghdr` structure followed by a data array. The data array contains the ancillary data message, and the `cmsghdr` structure contains descriptive information that allows an application to correctly parse the data.

The values for `msg_level` will be legal values for the level argument to the `getsockopt()` and `setsockopt()` functions. The `SCM_RIGHTS` type is supported for level `SOL_SOCKET`.

Ancillary data is also possible at the socket level. The `<sys/socket.h>` header defines the following macro for use as the `msg_type` value when `msg_level` is `SOL_SOCKET`:

`SCM_RIGHTS` Indicates that the data array contains the access rights to be sent or received.

The `<sys/socket.h>` header defines the following macros to gain access to the data arrays in the ancillary data associated with a message header:

`MSG_DATA(msg)` If the argument is a pointer to a `cmsghdr` structure, this macro returns an unsigned character pointer to the data array associated with the `cmsghdr` structure.

`MSG_NXTHDR(mhdr,msg)` If the first argument is a pointer to a `msg_hdr` structure and the second argument is a pointer to a `cmsghdr` structure in the ancillary data, pointed to by the `msg_control` field of that `msg_hdr` structure, this macro returns a pointer to the next `cmsghdr` structure, or a null pointer if this structure is the last `cmsghdr` in the ancillary data.

`MSG_FIRSTHDR(mhdr)` If the argument is a pointer to a `msg_hdr` structure, this macro returns a pointer to the first `cmsghdr` structure in the ancillary data associated with this `msg_hdr` structure, or a null pointer if there is no ancillary data associated with the `msg_hdr` structure.

The `<sys/socket.h>` header defines the `linger` structure that includes the following members:

int	l_onoff	/* indicates whether linger option is enabled */
int	l_linger	/* linger time, in seconds */

The `<sys/socket.h>` header defines the following macros:

SOCK_DGRAM	Datagram socket
SOCK_STREAM	Byte-stream socket
SOCK_SEQPACKET	Sequenced-packet socket

The `<sys/socket.h>` header defines the following macro for use as the *level* argument of `setsockopt()` and `getsockopt()`.

SOL_SOCKET Options to be accessed at socket level, not protocol level.

The `<sys/socket.h>` header defines the following macros: for use as the *option_name* argument in `getsockopt()` or `setsockopt()` calls:

SO_DEBUG	Debugging information is being recorded.
SO_ACCEPTCONN	Socket is accepting connections.
SO_BROADCAST	Transmission of broadcast messages is supported.
SO_REUSEADDR	Reuse of local addresses is supported.
SO_KEEPAIVE	Connections are kept alive with periodic messages.
SO_LINGER	Socket lingers on close.
SO_OOBINLINE	Out-of-band data is transmitted in line.
SO_SNDBUF	Send buffer size.
SO_RCVBUF	Receive buffer size.
SO_ERROR	Socket error status.
SO_TYPE	Socket type.

The `<sys/socket.h>` header defines the following macros for use as the valid values for the `msg_flags` field in the `msg_hdr` structure, or the `flags` parameter in `recvfrom()`, `recvmsg()`, `sendto()`, or `sendmsg()` calls:

MSG_TRUNC	Control data truncated.
MSG_EOR	Terminates a record (if supported by the protocol).
MSG_OOB	Out-of-band data.
MSG_PEEK	Leave received data in queue.

socket(3HEAD)

MSG_TRUNC Normal data truncated.

MSG_WAITALL Wait for complete message.

The `<sys/socket.h>` header defines the following macros:

AF_UNIX UNIX domain sockets

AF_INET Internet domain sockets

The `<sys/socket.h>` header defines the following macros:

SHUT_RD Disables further receive operations.

SHUT_WR Disables further send operations.

SHUT_RDWR Disables further send and receive operations.

The following are declared as functions, and may also be defined as macros:

```
int accept (int socket, struct sockaddr *address, size_t *address_len);
int bind (int socket, const struct sockaddr *address, size_t address_len);
int connect (int socket, const struct sockaddr *address, size_t address_len);
int getpeername (int socket, struct sockaddr *address, size_t *address_len);
int getsockname (int socket, struct sockaddr *address, size_t *address_len);
int getsockopt (int socket, int level, int option_name, void *option_value, size_t
*option_len);
int listen (int socket, int backlog);
ssize_t recv (int socket, void *buffer, size_t length, int flags);
ssize_t recvfrom (int socket, void *buffer, size_t length, int flags, struct sockaddr
*address, size_t *address_len);
ssize_t recvmsg (int socket, struct msghdr *message, int flags);
ssize_t send (int socket, const void *message, size_t length, int flags);
ssize_t sendmsg (int socket, const struct msghdr *message, int flags);
ssize_t sendto (int socket, const void *message, size_t length, int flags, const
struct sockaddr *dest_addr, size_t dest_len);
int setsockopt (int socket, int level, int option_name, const void *option_value,
size_t option_len);
int shutdown (int socket, int how);
```

socket(3HEAD)

```
int socket (int domain, int type, int protocol);
```

```
int socketpair (int domain, int type, int protocol, int socket_vector[2] );
```

SEE ALSO

```
accept(3SOCKET), accept(3XNET), bind(3SOCKET), bind(3XNET),  
connect(3SOCKET), connect(3XNET), getpeername(3SOCKET),  
getpeername(3XNET), getsockname(3SOCKET), getsockname(3XNET),  
getsockopt(3SOCKET), getsockopt(3XNET), listen(3SOCKET),  
listen(3XNET), recv(3SOCKET), recv(3XNET), recvfrom(3SOCKET),  
recvfrom(3XNET), recvmsg(3SOCKET), recvmsg(3XNET), send(3SOCKET),  
send(3XNET), sendmsg(3SOCKET), sendmsg(3XNET), sendto(3SOCKET),  
sendto(3XNET), setsockopt(3SOCKET), setsockopt(3XNET),  
shutdown(3SOCKET), shutdown(3XNET), socket(3SOCKET), socket(3XNET),  
socketpair(3SOCKET) socketpair(3XNET)
```

stat(3HEAD)

NAME	stat – data returned by stat system call
SYNOPSIS	<pre>#include <sys/types.h> #include <sys/stat.h></pre>
DESCRIPTION	<p>The system calls <code>stat</code>, <code>lstat</code> and <code>fstat</code> return data in a <code>stat</code> structure, which is defined in <code><stat.h></code>.</p> <p>The constants used in the <code>st_mode</code> field are also defined in this file:</p> <pre>#define S_IFMT /* type of file */ #define S_IAMB /* access mode bits */ #define S_IFIFO /* fifo */ #define S_IFCHR /* character special */ #define S_IFDIR /* directory */ #define S_IFNAM /* XENIX special named file */ #define S_INSEM /* XENIX semaphore subtype of IFNAM */ #define S_INSHD /* XENIX shared data subtype of IFNAM */ #define S_IFBLK /* block special */ #define S_IFREG /* regular */ #define S_IFLNK /* symbolic link */ #define S_IFSOCK /* socket */ #define S_IFDOOR /* door */ #define S_ISUID /* set user id on execution */ #define S_ISGID /* set group id on execution */ #define S_ISVTX /* save swapped text even after use */ #define S_IREAD /* read permission, owner */ #define S_IWRITE /* write permission, owner */ #define S_IEXEC /* execute/search permission, owner */ #define S_ENFMT /* record locking enforcement flag */ #define S_IRWXU /* read, write, execute: owner */ #define S_IRUSR /* read permission: owner */ #define S_IWUSR /* write permission: owner */ #define S_IXUSR /* execute permission: owner */</pre>


```
#define S_IRWXG /* read, write, execute: group */
#define S_IRGRP /* read permission: group */
#define S_IWGRP /* write permission: group */
#define S_IXGRP /* execute permission: group */
#define S_IRWXO /* read, write, execute: other */
#define S_IROTH /* read permission: other */
#define S_IWOTH /* write permission: other */
#define S_IXOTH /* execute permission: other */
```

The following macros are for POSIX conformance (see standards(5)):

```
#define S_ISBLK(mode) block special file
#define S_ISCHR(mode) character special file
#define S_ISDIR(mode) directory file
#define S_ISFIFO(mode) pipe or fifo file
#define S_ISREG(mode) regular file
#define S_ISSOCK(mode) socket file
```

SEE ALSO stat(2), standards(5), types(3HEAD)

stdarg(3HEAD)

NAME	stdarg – handle variable argument list
SYNOPSIS	<pre>#include <stdarg.h> va_list pvar; void va_start(va_list pvar, void parmN); (<i>type</i> *) va_arg(va_list pvar, <i>type</i>); void va_copy(va_list dest, va_list src); void va_end(va_list pvar);</pre>
DESCRIPTION	<p>This set of macros allows portable procedures that accept variable numbers of arguments of variable types to be written. Routines that have variable argument lists (such as <code>printf</code>) but do not use <code>stdarg</code> are inherently non-portable, as different machines use different argument-passing conventions.</p> <p><code>va_list</code> is a type defined for the variable used to traverse the list.</p> <p>The <code>va_start()</code> macro is invoked before any access to the unnamed arguments and initializes <code>pvar</code> for subsequent use by <code>va_arg()</code> and <code>va_end()</code>. The parameter <code>parmN</code> is the identifier of the rightmost parameter in the variable parameter list in the function definition (the one just before the <code>, ...</code>). If this parameter is declared with the <code>register</code> storage class or with a function or array type, or with a type that is not compatible with the type that results after application of the default argument promotions, the behavior is undefined.</p> <p>The parameter <code>parmN</code> is required under strict ANSI C compilation. In other compilation modes, <code>parmN</code> need not be supplied and the second parameter to the <code>va_start()</code> macro can be left empty (for example, <code>va_start(pvar,)</code>). This allows for routines that contain no parameters before the <code>...</code> in the variable parameter list.</p> <p>The <code>va_arg()</code> macro expands to an expression that has the type and value of the next argument in the call. The parameter <code>pvar</code> should have been previously initialized by <code>va_start()</code>. Each invocation of <code>va_arg()</code> modifies <code>pvar</code> so that the values of successive arguments are returned in turn. The parameter <code>type</code> is the type name of the next argument to be returned. The type name must be specified in such a way so that the type of a pointer to an object that has the specified type can be obtained simply by postfixing a <code>*</code> to <code>type</code>. If there is no actual next argument, or if <code>type</code> is not compatible with the type of the actual next argument (as promoted according to the default argument promotions), the behavior is undefined.</p> <p>The <code>va_copy()</code> macro saves the state represented by the <code>va_list src</code> in the <code>va_list dest</code>. The <code>va_list</code> passed as <code>dest</code> should not be initialized by a previous call to <code>va_start()</code>, and must be passed to <code>va_end()</code> before being reused as a parameter to <code>va_start()</code> or as the <code>dest</code> parameter of a subsequent call to <code>va_copy()</code>. The behavior is undefined should any of these restrictions not be met.</p> <p>The <code>va_end()</code> macro is used to clean up.</p>

Multiple traversals, each bracketed by `va_start` and `va_end`, are possible.

EXAMPLES **EXAMPLE 1** A sample program.

This example gathers into an array a list of arguments that are pointers to strings (but not more than `MAXARGS` arguments) with function `f1`, then passes the array as a single argument to function `f2`. The number of pointers is specified by the first argument to `f1`.

```
#include <stdarg.h>
#define MAXARGS 31
void f1(int n_ptrs, ...)
{
    va_list ap;
    char *array[MAXARGS];
    int ptr_no = 0;

    if (n_ptrs > MAXARGS)
        n_ptrs = MAXARGS;
    va_start(ap, n_ptrs);
    while (ptr_no < n_ptrs)
        array[ptr_no++] = va_arg(ap, char*);
    va_end(ap);
    f2(n_ptrs, array);
}
```

Each call to `f1` shall have visible the definition of the function or a declaration such as

```
void f1(int, ...)
```

SEE ALSO `vprintf(3C)`

NOTES It is up to the calling routine to specify in some manner how many arguments there are, since it is not always possible to determine the number of arguments from the stack frame. For example, `execl` is passed a zero pointer to signal the end of the list. `printf` can tell how many arguments there are by the format. It is non-portable to specify a second argument of `char`, `short`, or `float` to `va_arg`, because arguments seen by the called function are not `char`, `short`, or `float`. C converts `char` and `short` arguments to `int` and converts `float` arguments to `double` before passing them to a function.

time(3HEAD)

NAME	time – time types
SYNOPSIS	<pre>#include <time.h></pre>
DESCRIPTION	<p>The <code><time.h></code> header declares the structure <code>tm</code>, which includes the following members:</p> <pre>int tm_sec seconds [0,61] int tm_min minutes [0,59] int tm_hour hour [0,23] int tm_mday day of month [1,31] int tm_mon month of year [0,11] int tm_year years since 1900 int tm_wday day of week [0,6] (Sunday = 0) int tm_yday day of year [0,365] int tm_isdst daylight savings flag</pre> <p>The value of <code>tm_isdst</code> is positive if Daylight Saving Time is in effect, 0 if Daylight Saving Time is not in effect, and negative if the information is not available.</p> <p>This header defines the following symbolic names:</p> <p><code>NULL</code> Null pointer constant.</p> <p><code>CLK_TCK</code> Number of clock ticks per second returned by the <code>times(2)</code> function.</p> <p><code>CLOCKS_PER_SEC</code> A number used to convert the value returned by the <code>clock(3C)</code> function into seconds.</p> <p>The <code><time.h></code> header declares the structure <code>timespec</code>, which has the following members:</p> <pre>time_t tv_sec seconds long tv_nsec nanoseconds</pre> <p>This header also declares the <code>itimerspec</code> structure, which has at least the following members:</p> <pre>struct timespec it_interval timer period struct timespec it_value timer expiration</pre> <p>The following manifest constants are defined:</p> <p><code>CLOCK_REALTIME</code> The identifier of the systemwide realtime clock.</p> <p><code>TIMER_ABSTIME</code> Flag indicating time is absolute with respect to the clock associated with a timer.</p> <p>The <code>clock_t</code>, <code>size_t</code> and <code>time_t</code> types are defined as described in <code><sys/types.h></code>.</p> <p>Although the value of <code>CLOCKS_PER_SEC</code> is 1 million on all Solaris systems, it may be variable on other systems and it should not be assumed that <code>CLOCKS_PER_SEC</code> is a compile-time constant.</p>

time(3HEAD)

The value of `CLK_TCK` is currently the same as the value of `sysconf (_SC_CLK_TCK)`; however, new applications should call `sysconf(3C)` because the `CLK_TCK` macro may be withdrawn in a future issue.

The `<time.h>` header provides a declaration for `getdate_err`.

The following are declared as variables:

```
extern int      daylight;
extern long int timezone;
extern char     *tzname[ ];
```

USAGE The range [0,61] for `tm_sec` allows for the occasional leap second or double leap second.

`tm_year` is a signed value, therefore years before 1900 may be represented.

SEE ALSO `time(2)`, `times(2)`, `utime(2)`, `asctime(3C)`, `clock(3C)`, `clock_gettime(3RT)`, `ctime(3C)`, `difftime(3C)`, `getdate(3C)`, `gmtime(3C)`, `localtime(3C)`, `mktime(3C)`, `nanosleep(3RT)`, `strftime(3C)`, `strptime(3C)`, `sysconf(3C)`, `timer_create(3RT)`, `timer_delete(3RT)`, `timer_gettime(3RT)`, `tzset(3C)`

types32(3HEAD)

NAME	types32 – fixed-width data types
SYNOPSIS	<pre>#include <sys/types32.h></pre>
DESCRIPTION	The following fixed-width data types defined in <code><sys/types32.h></code> correspond to the sign and sizes of types in the 32-bit environment that can be used for compatibility and interoperability purposes in either the 32-bit or 64-bit environment.

typedef	int32_t	blkcnt32_t
typedef	uint32_t	caddr32_t
typedef	int32_t	clock32_t
typedef	int32_t	daddr32_t
typedef	uint32_t	dev32_t
typedef	uint32_t	fsblkcnt32_t
typedef	uint32_t	fsfilcnt32_t
typedef	int32_t	gid32_t
typedef	int32_t	id32_t
typedef	uint32_t	ino32_t
typedef	int32_t	key32_t
typedef	uint32_t	major32_t
typedef	uint32_t	minor32_t
typedef	uint32_t	mode32_t
typedef	uint32_t	nlink32_t
typedef	int32_t	pid32_t
typedef	uint32_t	rlim32_t
typedef	uint32_t	size32_t
typedef	int32_t	ssize32_t
typedef	time32_t	int32_t
typedef	uid32_t	int32_t

NAME	types – primitive system data types
SYNOPSIS	#include <sys/types.h>
DESCRIPTION	The data types defined in <sys/types.h> are discussed.
32-bit Solaris	<p>The data types listed below are defined in <sys/types.h> for 32-bit Solaris.</p> <pre> typedef struct { int r[1]; } *physadr; typedef long clock_t; typedef long daddr_t; typedef char * caddr_t; typedef unsigned char unchar; typedef unsigned short ushort; typedef unsigned int uint; typedef unsigned long ulong_t; typedef unsigned long ino_t; typedef long uid_t; typedef long gid_t; typedef ulong_t nlink_t; typedef ulong_t mode_t; typedef short cnt_t; typedef long time_t; typedef int label_t[10]; typedef ulong_t dev_t; typedef long off_t; typedef long pid_t; typedef long paddr_t; typedef int key_t; typedef unsigned char use_t; typedef short sysid_t; typedef short index_t; typedef short lock_t; typedef unsigned int size_t; typedef long clock_t; typedef long pid_t; </pre>
64-bit Solaris	<p>The data types listed below are defined in <sys/types.h> for 64-bit Solaris.</p> <pre> typedef long blkcnt_t typedef long clock_t typedef long daddr_t typedef ulong_t dev_t typedef ulong_t fsblkcnt_t typedef ulong_t fsfilcnt_t typedef int gid_t typedef int id_t typedef long ino_t typedef int key_t typedef uint_t major_t typedef uint_t minor_t typedef uint_t mode_t typedef uint_t nlink_t typedef int pid_t typedef ptrdiff_t intptr_t typedef ulong_t rlim_t typedef ulong_t size_t typedef uint_t speed_t </pre>

types(3HEAD)

```
typedef    long        ssize_t
typedef    long        suseconds_t
typedef    uint_t      tcflag_t
typedef    long        time_t
typedef    int         uid_t
typedef    int         wchar_t
```

USAGE The `daddr_t` type is used for disk addresses except in an inode on disk. Times are encoded in seconds since 00:00:00 UTC, January 1, 1970. The major and minor parts of a device code specify kind and unit number of a device and are installation-dependent. Offsets are measured in bytes from the beginning of a file.

The `label_t []` types are used to save the processor state while another process is running.

NOTES For 32-bit programs, pointers and the C data types `int` and `long` are all 32-bit quantities. For 64-bit programs, pointers and the C data type `long` are defined as 64-bit quantities.

The preprocessor symbol `_ILP32`, made visible by the inclusion of `<sys/types.h>` can be used with the preprocessor `#ifdef` construct to define sections of code that will *only* be compiled as part of a 32-bit version of a given C program.

The preprocessor symbol `_LP64` can be used in the same way to define sections of code that will *only* be compiled as part of a 64-bit version of a given C program.

For example:

```
#include <sys/types.h>
...

#ifdef _LP64
    printf("The data model is LP64 in this environment\n");
#else
#ifdef _ILP32
    printf("The data model is ILP32 in this environment\n");
#else
#error    "Unknown data model!"
#endif
#endif
```


NAME	ucontext – user context
SYNOPSIS	<code>#include <ucontext.h></code>
DESCRIPTION	<p>The <code>ucontext</code> structure defines the context of a thread of control within an executing process.</p> <p>This structure includes at least the following members:</p> <pre>ucontext_t uc_link sigset_t uc_sigmask stack_t uc_stack mcontext_t uc_mcontext</pre> <p><code>uc_link</code> is a pointer to the context that to be resumed when this context returns. If <code>uc_link</code> is equal to 0, then this context is the main context, and the process exits when this context returns.</p> <p><code>uc_sigmask</code> defines the set of signals that are blocked when this context is active [see <code>sigprocmask(2)</code>].</p> <p><code>uc_stack</code> defines the stack used by this context [see <code>sigaltstack(2)</code>].</p> <p><code>uc_mcontext</code> contains the saved set of machine registers and any implementation specific context data. Portable applications should not modify or access <code>uc_mcontext</code>.</p>
SEE ALSO	<code>getcontext(2)</code> , <code>sigaction(2)</code> , <code>sigaltstack(2)</code> , <code>sigprocmask(2)</code> , <code>makecontext(3C)</code>

un(3HEAD)

NAME	un – definitions for UNIX-domain sockets
SYNOPSIS	<pre>#include <sys/un.h></pre>
DESCRIPTION	<p>The <code><sys/un.h></code> header defines the <code>sockaddr_un</code> structure that includes the following members:</p> <pre>sa_family_t sun_family /* address family */ char sun_path[] /* socket pathname */</pre> <p>The <code>sockaddr_un</code> structure is used to store addresses for UNIX domain sockets. Values of this type must be cast to <code>struct sockaddr</code> for use with the socket interfaces.</p> <p>The <code><sys/un.h></code> header defines the type <code>sa_family_t</code> as described in <code>socket(3HEAD)</code>.</p>
SEE ALSO	<code>bind(3SOCKET)</code> , <code>bind(3XNET)</code> , <code>socket(3HEAD)</code> , <code>socket(3SOCKET)</code> , <code>socket(3XNET)</code> , <code>socketpair(3SOCKET)</code> , <code>socketpair(3XNET)</code>

NAME	unistd – standard symbolic constants and types																
SYNOPSIS	<code>#include <unistd.h></code>																
DESCRIPTION	The <code><unistd.h></code> 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.																
Version Test Macros	<p>The following symbolic constants are defined (with fixed values):</p> <table border="0"> <tr> <td style="padding-right: 20px;"><code>_POSIX_VERSION</code></td> <td>Integer value indicating version of the POSIX standard (C language binding). See <code>standards(5)</code>.</td> </tr> <tr> <td><code>_POSIX2_VERSION</code></td> <td>Integer value indicating version of the POSIX.2 standard (Commands). <code>_POSIX2_C_VERSION</code> Integer value indicating version of the POSIX.2 standard (C language binding).</td> </tr> <tr> <td><code>_XOPEN_VERSION</code></td> <td>Integer value indicating version of the XPG to which system conforms.</td> </tr> <tr> <td><code>_XOPEN_XCU_VERSION</code></td> <td>Integer value indicating the version of the XCU specification to which the implementation conforms. If this constant is not defined, use the <code>sysconf(3C)</code> function to determine which features are supported.</td> </tr> </table>	<code>_POSIX_VERSION</code>	Integer value indicating version of the POSIX standard (C language binding). See <code>standards(5)</code> .	<code>_POSIX2_VERSION</code>	Integer value indicating version of the POSIX.2 standard (Commands). <code>_POSIX2_C_VERSION</code> Integer value indicating version of the POSIX.2 standard (C language binding).	<code>_XOPEN_VERSION</code>	Integer value indicating version of the XPG to which system conforms.	<code>_XOPEN_XCU_VERSION</code>	Integer value indicating the version of the XCU specification to which the implementation conforms. If this constant is not defined, use the <code>sysconf(3C)</code> function to determine which features are supported.								
<code>_POSIX_VERSION</code>	Integer value indicating version of the POSIX standard (C language binding). See <code>standards(5)</code> .																
<code>_POSIX2_VERSION</code>	Integer value indicating version of the POSIX.2 standard (Commands). <code>_POSIX2_C_VERSION</code> Integer value indicating version of the POSIX.2 standard (C language binding).																
<code>_XOPEN_VERSION</code>	Integer value indicating version of the XPG to which system conforms.																
<code>_XOPEN_XCU_VERSION</code>	Integer value indicating the version of the XCU specification to which the implementation conforms. If this constant is not defined, use the <code>sysconf(3C)</code> function to determine which features are supported.																
Mandatory Symbolic Constants	<p>The following symbolic constants are either undefined or defined with a value other than <code>-1</code>. If a constant is undefined, an application should use the <code>sysconf(3C)</code>, <code>pathconf(2)</code>, or <code>fpathconf(2)</code> functions to determine which features are present on the system at that time or for the particular pathname in question.</p> <table border="0"> <tr> <td style="padding-right: 20px;"><code>_POSIX_JOB_CONTROL</code></td> <td>Implementation supports job control.</td> </tr> <tr> <td><code>_POSIX_SAVED_IDS</code></td> <td>The <code>exec</code> functions (see <code>exec(2)</code>) save the effective user and group.</td> </tr> <tr> <td><code>_POSIX_THREADS</code></td> <td>The implementation supports the threads option.</td> </tr> <tr> <td><code>_POSIX_THREAD_ATTR_STACKADDR</code></td> <td>The implementation supports the thread stack address attribute option.</td> </tr> <tr> <td><code>_POSIX_THREAD_ATTR_STACKSIZE</code></td> <td>The implementation supports the thread stack size attribute option.</td> </tr> <tr> <td><code>_POSIX_THREAD_PROCESS_SHARED</code></td> <td>The implementation supports the process-shared synchronization option.</td> </tr> <tr> <td><code>_POSIX_THREAD_SAFE_FUNCTIONS</code></td> <td>The implementation supports the thread-safe functions option.</td> </tr> <tr> <td><code>_XOPEN_XPG3</code></td> <td>X/Open Specification, February 1992, System Interfaces and Headers, Issue 3 (ISBN: 1-872630-37-5, C212); this</td> </tr> </table>	<code>_POSIX_JOB_CONTROL</code>	Implementation supports job control.	<code>_POSIX_SAVED_IDS</code>	The <code>exec</code> functions (see <code>exec(2)</code>) save the effective user and group.	<code>_POSIX_THREADS</code>	The implementation supports the threads option.	<code>_POSIX_THREAD_ATTR_STACKADDR</code>	The implementation supports the thread stack address attribute option.	<code>_POSIX_THREAD_ATTR_STACKSIZE</code>	The implementation supports the thread stack size attribute option.	<code>_POSIX_THREAD_PROCESS_SHARED</code>	The implementation supports the process-shared synchronization option.	<code>_POSIX_THREAD_SAFE_FUNCTIONS</code>	The implementation supports the thread-safe functions option.	<code>_XOPEN_XPG3</code>	X/Open Specification, February 1992, System Interfaces and Headers, Issue 3 (ISBN: 1-872630-37-5, C212); this
<code>_POSIX_JOB_CONTROL</code>	Implementation supports job control.																
<code>_POSIX_SAVED_IDS</code>	The <code>exec</code> functions (see <code>exec(2)</code>) save the effective user and group.																
<code>_POSIX_THREADS</code>	The implementation supports the threads option.																
<code>_POSIX_THREAD_ATTR_STACKADDR</code>	The implementation supports the thread stack address attribute option.																
<code>_POSIX_THREAD_ATTR_STACKSIZE</code>	The implementation supports the thread stack size attribute option.																
<code>_POSIX_THREAD_PROCESS_SHARED</code>	The implementation supports the process-shared synchronization option.																
<code>_POSIX_THREAD_SAFE_FUNCTIONS</code>	The implementation supports the thread-safe functions option.																
<code>_XOPEN_XPG3</code>	X/Open Specification, February 1992, System Interfaces and Headers, Issue 3 (ISBN: 1-872630-37-5, C212); this																

unistd(3HEAD)

**Constants for
Options and
Feature Groups**

specification was formerly X/Open Portability Guide, Issue 3, Volume 2, January 1989, XSI System Interface and Headers (ISBN: 0-13-685843-0, XO/XPG/89/003).

`_XOPEN_XPG4` X/Open CAE Specification, July 1992, System Interfaces and Headers, Issue 4 (ISBN: 1-872630-47-2, C202).

`_XOPEN_UNIX` X/Open CAE Specification, January 1997, System Interfaces and Headers, Issue 5 (ISBN: 1-85912-181-0, C606).

The following symbolic constants are defined to have the value `-1` if the implementation will never provide the feature, and to have a value other than `-1` if the implementation always provides the feature. If these are undefined, the `sysconf()` function can be used to determine whether the feature is provided for a particular invocation of the application.

`_POSIX2_C_BIND`
Implementation supports the C Language Binding option.

`_POSIX2_C_DEV`
Implementation supports the C Language Development Utilities option.

`_POSIX2_CHAR_TERM`
Implementation supports at least one terminal type.

`_POSIX2_LOCALEDEF`
Implementation supports the creation of locales by the `localedef(1)` utility.

`_POSIX2_SW_DEV`
Implementation supports the Software Development Utilities option.

`_POSIX2_UPE`
The implementation supports the User Portability Utilities option.

`_XOPEN_ENH_I18N`
The implementation supports the Issue 4, Version 2 Enhanced Internationalization Feature Group.

`_XOPEN_LEGACY`
The implementation supports the Legacy Feature Group.

`_XOPEN_REALTIME`
The implementation supports the X/Open Realtime Feature Group.

`_XOPEN_SHM`
The implementation supports the Issue 4, Version 2 Shared Memory Feature Group.

`_XBS5_ILP32_OFF32`
Implementation provides a C-language compilation environment with 32-bit `int`, `long`, `pointer` and `off_t` types.

`_XBS5_ILP32_OFFBIG`

Implementation provides a C-language compilation environment with 32-bit `int`, `long` and `pointer` types and an `off_t` type using at least 64 bits.

`_XBS5_LP64_OFF64`

Implementation provides a C-language compilation environment with 32-bit `int` and 64-bit `long`, `pointer` and `off_t` types.

`_XBS5_LPBIG_OFFBIG`

Implementation provides a C-language compilation environment with an `int` type using at least 32 bits and `long`, `pointer` and `off_t` types using at least 64 bits.

If `_XOPEN_REALTIME` is defined to have a value other than `-1` then the following symbolic constants will be defined to an unspecified value to indicate that the features are supported.

<code>_POSIX_ASYNCHRONOUS_IO</code>	Implementation supports the Asynchronous Input and Output option.
<code>_POSIX_MEMLOCK</code>	Implementation supports the Process Memory Locking option.
<code>_POSIX_MEMLOCK_RANGE</code>	Implementation supports the Range Memory Locking option.
<code>_POSIX_MESSAGE_PASSING</code>	Implementation supports the Message Passing option.
<code>_POSIX_PRIORITY_SCHEDULING</code>	Implementation supports the Process Scheduling option.
<code>_POSIX_REALTIME_SIGNALS</code>	Implementation supports the Realtime Signals Extension option.
<code>_POSIX_SEMAPHORES</code>	Implementation supports the Semaphores option.
<code>_POSIX_SHARED_MEMORY_OBJECTS</code>	Implementation supports the Shared Memory Objects option.
<code>_POSIX_SYNCHRONIZED_IO</code>	Implementation supports the Synchronized Input and Output option.
<code>_POSIX_TIMERS</code>	Implementation supports the Timers option.

The following symbolic constants are always defined to unspecified values to indicate that the functionality is always present on XSI-conformant systems.

<code>_POSIX_FSYNC</code>	Implementation supports the File Synchronisation option.
<code>_POSIX_MAPPED_FILES</code>	Implementation supports the Memory Mapped Files option.

unistd(3HEAD)

	<code>_POSIX_MEMORY_PROTECTION</code>	Implementation supports the Memory Protection option.
Execution-time Symbolic Constants	If any of the following constants are not defined in the header <code><unistd.h></code> , the value varies depending on the file to which it is applied.	
	If any of the following constants are defined to have value <code>-1</code> in the header <code><unistd.h></code> , the implementation will not provide the option on any file; if any are defined to have a value other than <code>-1</code> in the header <code><unistd.h></code> , the implementation will provide the option on all applicable files.	
	All of the following constants, whether defined in <code><unistd.h></code> or not, may be queried with respect to a specific file using the <code>pathconf()</code> or <code>fpathconf()</code> functions.	
	<code>_POSIX_ASYNC_IO</code>	Asynchronous input or output operations may be performed for the associated file.
	<code>_POSIX_PRIO_IO</code>	Prioritized input or output operations may be performed for the associated file.
	<code>_POSIX_SYNC_IO</code>	Synchronized input or output operations may be performed for the associated file.
Constants for Functions	The following constant is defined:	
	<code>NULL</code>	Null pointer.
	The following symbolic constants are defined for the <code>access(2)</code> function:	
	<code>R_OK</code>	Test for read permission.
	<code>W_OK</code>	Test for write permission.
	<code>X_OK</code>	Test for execute (search) permission.
	<code>F_OK</code>	Test for existence of file. The constants <code>F_OK</code> , <code>R_OK</code> , <code>W_OK</code> , and <code>X_OK</code> , and the expressions <code>R_OK W_OK</code> , <code>R_OK X_OK</code> , and <code>R_OK W_OK X_OK</code> all have distinct values.
	The following symbolic constants are defined for the <code>lockf(3C)</code> function:	
	<code>F_ULOCK</code>	Unlock a previously locked region.
	<code>F_LOCK</code>	Lock a region for exclusive use.
	<code>F_TLOCK</code>	Test and lock a region for exclusive use.
	<code>F_TEST</code>	Test a region for other processes locks.
	The following symbolic constants are defined for the <code>lseek(2)</code> and <code>fcntl(2)</code> functions (they have distinct values):	
	<code>SEEK_SET</code>	Set file offset to <i>offset</i> .

SEEK_CUR Set file offset to current plus *offset*.

SEEK_END Set file offset to EOF plus *offset*.

The following symbolic constants are defined for the `confstr(3C)` function for both SPARC and x86:

_CS_LFS64_CFLAGS	_CS_LFS64_LDFLAGS
_CS_LFS64_LIBS	_CS_LFS64_LINTFLAGS
_CS_LFS_CFLAGS	_CS_LFS_LDFLAGS
_CS_LFS_LIBS	_CS_LFS_LINTFLAGS
_CS_PATH	_CS_XBS5_ILP32_OFF32_CFLAGS
_CS_XBS5_ILP32_OFF32_LDFLAGS	_CS_XBS5_ILP32_OFF32_LIBS
_CS_XBS5_ILP32_OFF32_LINTFLAGS	_CS_XBS5_ILP32_OFFBIG_CFLAGS
_CS_XBS5_ILP32_OFFBIG_LDFLAGS	_CS_XBS5_ILP32_OFFBIG_LIBS
_CS_XBS5_ILP32_OFFBIG_LINTFLAGS	

The following symbolic constants are defined for the `confstr()` function for SPARC only:

_CS_XBS5_LP64_OFF64_CFLAGS	_CS_XBS5_LP64_OFF64_LDFLAGS
_CS_XBS5_LP64_OFF64_LIBS	_CS_XBS5_LP64_OFF64_LINTFLAGS
_CS_XBS5_LPBIG_OFFBIG_CFLAGS	_CS_XBS5_LPBIG_OFFBIG_LDFLAGS
_CS_XBS5_LPBIG_OFFBIG_LIBS	_CS_XBS5_LPBIG_OFFBIG_LINTFLAGS

The following symbolic constants are defined for the `sysconf(3C)` function:

_SC_2_C_BIND	_SC_2_C_DEV
_SC_2_C_VERSION	_SC_2_FORT_DEV
_SC_2_FORT_RUN	_SC_2_LOCALEDEF
_SC_2_SW_DEV	_SC_2_UPE
_SC_2_VERSION	_SC_AIO_LISTIO_MAX
_SC_AIO_MAX	_SC_AIO_PRIO_DELTA_MAX
_SC_ARG_MAX	_SC_ASYNCHRONOUS_IO

unistd(3HEAD)

_SC_ATEXIT_MAX	_SC_AVPHYS_PAGES
_SC_BC_BASE_MAX	_SC_BC_DIM_MAX
_SC_BC_SCALE_MAX	_SC_BC_STRING_MAX
_SC_CHILD_MAX	_SC_CLK_TCK
_SC_COLL_WEIGHTS_MAX	_SC_DELAYTIMER_MAX
_SC_EXPR_NEST_MAX	_SC_FSYNC
_SC_GETGR_R_SIZE_MAX	_SC_GETPW_R_SIZE_MAX
_SC_IOV_MAX	_SC_JOB_CONTROL
_SC_LINE_MAX	_SC_LOGIN_NAME_MAX
_SC_LOGNAME_MAX	_SC_MAPPED_FILES
_SC_MEMLOCK	_SC_MEMLOCK_RANGE
_SC_MEMORY_PROTECTION	_SC_MESSAGE_PASSING
_SC_MQ_OPEN_MAX	_SC_MQ_PRIO_MAX
_SC_NGROUPS_MAX	_SC_NPROCESSORS_CONF
_SC_NPROCESSORS_ONLN	_SC_OPEN_MAX
_SC_PAGESIZE	_SC_PAGE_SIZE
_SC_PASS_MAX	_SC_PHYS_PAGES
_SC_PRIORITIZED_IO	_SC_PRIORITY_SCHEDULING
_SC_REALTIME_SIGNALS	_SC_RE_DUP_MAX
_SC_RTSIG_MAX	_SC_SAVED_IDS
_SC_SEMAPHORES	_SC_SEM_NSEMS_MAX
_SC_SEM_VALUE_MAX	_SC_SHARED_MEMORY_OBJECTS
_SC_SIGQUEUE_MAX	_SC_STREAM_MAX
_SC_SYNCHRONIZED_IO	_SC_THREAD_ATTR_STACKADDR
_SC_THREAD_ATTR_STACKSIZE	_SC_THREAD_DESTRUCTOR_ITERATIONS
_SC_THREAD_KEYS_MAX	_SC_THREAD_PRIO_INHERIT
_SC_THREAD_PRIO_PROTECT	_SC_THREAD_PRIORITY_SCHEDULING
_SC_THREAD_PROCESS_SHARED	_SC_THREADS
_SC_THREAD_SAFE_FUNCTIONS	_SC_THREAD_STACK_MIN
_SC_THREAD_THREADS_MAX	_SC_TIMER_MAX

<code>_SC_TIMERS</code>	<code>_SC_TTY_NAME_MAX</code>
<code>_SC_TZNAME_MAX</code>	<code>_SC_VERSION</code>
<code>_SC_XBS5_ILP32_OFF32</code>	<code>_SC_XBS5_ILP32_OFFBIG</code>
<code>_SC_XBS5_LP64_OFF64</code>	<code>_SC_XBS5_LPBIG_OFFBIG</code>
<code>_SC_XOPEN_CRYPT</code>	<code>_SC_XOPEN_ENH_I18N</code>
<code>_SC_XOPEN_SHM</code>	<code>_SC_XOPEN_UNIX</code>
<code>_SC_XOPEN_VERSION</code>	<code>_SC_XOPEN_XCU_VERSION</code>

The two constants `_SC_PAGESIZE` and `_SC_PAGE_SIZE` may be defined to have the same value.

The following symbolic constants are defined for the `fpathconf(2)` function:

<code>_PC_ASYNC_IO</code>	<code>_PC_CHOWN_RESTRICTED</code>
<code>_PC_FILESIZEBITS</code>	<code>_PC_LINK_MAX</code>
<code>_PC_MAX_CANON</code>	<code>_PC_MAX_INPUT</code>
<code>_PC_NAME_MAX</code>	<code>_PC_NO_TRUNC</code>
<code>_PC_PATH_MAX</code>	<code>_PC_PIPE_BUF</code>
<code>_PC_PRIO_IO</code>	<code>_PC_SYNC_IO</code>
<code>_PC_VDISABLE</code>	<code>_PC_XATTR_ENABLED</code>
<code>_PC_XATTR_EXISTS</code>	

The following symbolic constants are defined for file streams:

<code>STDIN_FILENO</code>	File number (0) of <code>stdin</code> .
<code>STDOUT_FILENO</code>	File number (1) of <code>stdout</code> .
<code>STDERR_FILENO</code>	File number (2) of <code>stderr</code> . The following pathnames are defined:
<code>GF_PATH</code>	Pathname of the group file.
<code>PF_PATH</code>	Pathname of the passwd file.

SEE ALSO `access(2)`, `exec(2)`, `fcntl(2)`, `fpathconf(2)`, `lseek(2)`, `confstr(3C)`, `lockf(3C)`, `sysconf(3C)`, `termios(3C)`, `group(4)`, `passwd(4)`, `standards(5)`, `termio(7I)`

values(3HEAD)

NAME	values – machine-dependent values
SYNOPSIS	<code>#include <values.h></code>
DESCRIPTION	<p>This file contains a set of manifest constants, conditionally defined for particular processor architectures.</p> <p>The model assumed for integers is binary representation (one's or two's complement), where the sign is represented by the value of the high-order bit.</p> <p>BITS (type) The number of bits in a specified type (for example, <code>int</code>).</p> <p>HIBITS The value of a short integer with only the high-order bit set.</p> <p>HIBITL The value of a long integer with only the high-order bit set.</p> <p>HIBITI The value of a regular integer with only the high-order bit set.</p> <p>MAXSHORT The maximum value of a signed short integer.</p> <p>MAXLONG The maximum value of a signed long integer.</p> <p>MAXINT The maximum value of a signed regular integer.</p> <p>MAXFLOAT, LN_MAXFLOAT The maximum value of a single-precision floating-point number, and its natural logarithm.</p> <p>MAXDOUBLE, LN_MAXDOUBLE The maximum value of a double-precision floating-point number, and its natural logarithm.</p> <p>MINFLOAT, LN_MINFLOAT The minimum positive value of a single-precision floating-point number, and its natural logarithm.</p> <p>MINDOUBLE, LN_MINDOUBLE The minimum positive value of a double-precision floating-point number, and its natural logarithm.</p> <p>FSIGNIF The number of significant bits in the mantissa of a single-precision floating-point number.</p> <p>DSIGNIF The number of significant bits in the mantissa of a double-precision floating-point number.</p>

values(3HEAD)

SEE ALSO | intro(3) math(3HEAD)

varargs(3HEAD)

NAME	varargs – handle variable argument list
SYNOPSIS	<pre>#include <varargs.h> va_alist va_dcl va_list pvar; void va_start(va_list pvar); type va_arg(va_list pvar, type); void va_end(va_list pvar);</pre>
DESCRIPTION	<p>This set of macros allows portable procedures that accept variable argument lists to be written. Routines that have variable argument lists (such as <code>printf(3C)</code>) but do not use <code>varargs</code> are inherently non-portable, as different machines use different argument-passing conventions.</p> <p><code>va_alist</code> is used as the parameter list in a function header.</p> <p><code>va_dcl</code> is a declaration for <code>va_alist</code>. No semicolon should follow <code>va_dcl</code>.</p> <p><code>va_list</code> is a type defined for the variable used to traverse the list.</p> <p><code>va_start</code> is called to initialize <code>pvar</code> to the beginning of the list.</p> <p><code>va_arg</code> will return the next argument in the list pointed to by <code>pvar</code>. <code>type</code> is the type the argument is expected to be. Different types can be mixed, but it is up to the routine to know what type of argument is expected, as it cannot be determined at runtime.</p> <p><code>va_end</code> is used to clean up.</p> <p>Multiple traversals, each bracketed by <code>va_start</code> and <code>va_end</code>, are possible.</p>
EXAMPLES	<p>EXAMPLE 1 A sample program.</p> <p>This example is a possible implementation of <code>execl</code> (see <code>exec(2)</code>).</p> <pre>#include <unistd.h> #include <varargs.h> #define MAXARGS 100 /* execl is called by execl(file, arg1, arg2, ..., (char *)0); */ execl(va_alist) va_dcl { va_list ap; char *file; char *args[MAXARGS]; /* assumed big enough*/ int argno = 0; va_start(ap); file = va_arg(ap, char *); while ((args[argno++] = va_arg(ap, char *)) != 0)</pre>

EXAMPLE 1 A sample program. (Continued)

```
    ;  
    va_end(ap);  
    return execv(file, args);  
}
```

SEE ALSO `exec(2)`, `printf(3C)`, `vprintf(3C)`, `stdarg(3HEAD)`

NOTES It is up to the calling routine to specify in some manner how many arguments there are, since it is not always possible to determine the number of arguments from the stack frame. For example, `execl` is passed a zero pointer to signal the end of the list. `printf` can tell how many arguments are there by the format.

It is non-portable to specify a second argument of `char`, `short`, or `float` to `va_arg`, since arguments seen by the called function are not `char`, `short`, or `float`. C converts `char` and `short` arguments to `int` and converts `float` arguments to `double` before passing them to a function.

`stdarg` is the preferred interface.

wstat(3HEAD)

NAME	wstat – wait status
SYNOPSIS	#include <sys/wait.h>
DESCRIPTION	<p>When a process waits for status from its children via either the <code>wait</code> or <code>waitpid</code> function, the status returned may be evaluated with the following macros, defined in <sys/wait.h>. These macros evaluate to integral expressions. The <i>stat</i> argument to these macros is the integer value returned from <code>wait</code> or <code>waitpid</code>.</p> <p>WIFEXITED (<i>stat</i>) Evaluates to a non-zero value if status was returned for a child process that terminated normally.</p> <p>WEXITSTATUS (<i>stat</i>) If the value of <code>WIFEXITED (<i>stat</i>)</code> is non-zero, this macro evaluates to the exit code that the child process passed to <code>_exit()</code> (see <code>exit(2)</code>) or <code>exit(3C)</code>, or the value that the child process returned from <code>main</code>.</p> <p>WIFSIGNALED (<i>stat</i>) Evaluates to a non-zero value if status was returned for a child process that terminated due to the receipt of a signal.</p> <p>WTERMSIG (<i>stat</i>) If the value of <code>WIFSIGNALED (<i>stat</i>)</code> is non-zero, this macro evaluates to the number of the signal that caused the termination of the child process.</p> <p>WIFSTOPPED (<i>stat</i>) Evaluates to a non-zero value if status was returned for a child process that is currently stopped.</p> <p>WSTOPSIG (<i>stat</i>) If the value of <code>WIFSTOPPED (<i>stat</i>)</code> is non-zero, this macro evaluates to the number of the signal that caused the child process to stop.</p> <p>WIFCONTINUED(<i>stat</i>) Evaluates to a non-zero value if status was returned for a child process that has continued.</p> <p>WCOREDUMP(<i>stat</i>) If the value of <code>WIFSIGNALED (<i>stat</i>)</code> is non-zero, this macro evaluates to a non-zero value if a core image of the terminated child was created.</p>
SEE ALSO	<code>exit(2)</code> , <code>wait(2)</code> , <code>waitpid(2)</code> , <code>exit(3C)</code>

Index

A

accounting files, — acct, 28
acct — process accounting file format, 28
alternative memory allocator library —
libmapmalloc, 121
ar — archive file format, 31
archive file format — ar, 31
asynchronous I/O library — libaio, 48

B

basic security library — libbsm, 50

C

C library — libc, 52
configuration adminstartion library —
libcfgadm, 79

D

data types, primitive system, — types, 239
definitions for internet operations — inet, 43
definitions for ndbm database operations —
ndbm, 209
definitions for network database operations —
netdb, 210
definitions for UNIX-domain sockets — un, 242
device ID library — libdevid, 92
layout service library — liblayout, 117

name—value pair library — libnvpair, 135
device information library — libdevinfo, 93
dirent — file system independent directory
entry, 34
dynamic linking library — libdl, 95

E

ELF access library — libelf, 100
encryption/decryption library — libcrypt, 81

F

File Access Control List library — libsec, 164
file control options, — fcntl, 35
fixed-width data types — types32, 238
floatingpoint — IEEE floating point
definitions, 39
forms library — libform, 104

G

general administrative library — libadm, 47
Generic Security Services library — libgss, 108
graphics interface libraries
— lib300, 144
— lib300s, 144
— lib4014, 144
— lib450, 144
— libplot, 144

graphics interface libraries (Continued)
— libvt0, 144

I

IEEE arithmetic, floating point definitions —
floatingpoint, 39
in — Internet Protocol family, 41
inet — definitions for internet operations, 43
Default, 43
Standard-conforming, 43
internationalization library — libintl, 113
Internet Protocol family — in, 41
Internet Protocol family — socket, 227

K

kernel statistics library — libkstat, 114
Kernel Virtual Memory access library —
libkvm, 115

L

langinfo — language information constants, 44
language data types, native — nl_types, 214
language information constants — langinfo, 44
lex library — libl, 116
lib300 — graphics interface libraries, 144
lib300s — graphics interface libraries, 144
lib4014 — graphics interface libraries, 144
lib450 — graphics interface libraries, 144
libadm — general administrative library, 47
libaio — asynchronous I/O library, 48
libbsdmalloc — memory allocator interface
library, 49
libbsm — basic security library, 50
libc — C library, 52
libc — the C library, 52
libcfgadm — configuration adminstartion
library, 79
libcrypt — encryption/decryption library, 81
libcurses — screen handling and optimization
library, 82
libdevic — device ID library, 92
liblayout — layout service library, 117

libnvpair — name—value pair library, 135
libdevinfo — device information library, 93
libdl — dynamic linking library, 95
libdl — the dynamic linking interface
library, 95
libdmi — Sun Solstice Enterprise Agent DMI
library, 96
libdmici — Sun Solstice Enterprise Agent
Component library, 97
libdmimi — Sun Solstice Enterprise Agent
Management library, 98
libelf — ELF access library, 100
libform — forms library, 104
libgen — string pattern-matching library, 106
libgss — Generic Security Services library, 108
libintl — internationalization library, 113
libkstat — kernel statistics library, 114
libkvm — Kernel Virtual Memory access
library, 115
/usr/lib/libkvm.so.1, 115
libl — lex library, 116
libmail — user mailbox lockfile management
library, 119
libmalloc — memory allocation library, 120
libmapmalloc — alternative memory allocator
library, 121
libmenu — menus library, 123
libmp — multiple precision library, 125
libmtmalloc — multi-threaded memory
allocator library, 126
libnsl — network services library, 127
libpam — PAM (Pluggable Authentication
Module) library, 137
libpanel — panels library, 139
libpicl — PICL library, 141
libpicltree — PICL plug-in library, 142
libplot — graphics interface libraries, 144
libpool — pool configuration manipulation
library, 146
libposix4 — POSIX.1b Realtime Extensions
library, 161
libpthread — POSIX threads library, 151
libpthread — posix threads library,
/usr/lib/libpthread.so.1, 151
librac — remote asynchronous calls library, 154
library file format — ar, 31
libresolv — resolver library, 155
librpcsoc — obsolete RPC library, 157

librpcsvc — RPC services library, 158
 librsn — remote shared memory interface library, 159
 librt — POSIX.1b Realtime Extensions library, 161
 librtld_db — runtime linker debugging library, 163
 libsec — File Access Control List library, 164
 /usr/lib/libsec.so.1, 164
 libsecdb — security attributes database library, 165
 libsendfile — sendfile library, 167
 libslp — service location protocol library, 168
 libsmartcard — smartcard library, 169
 libsocket — sockets library, 171
 libssagent — Sun Solstice Enterprise Agent library, 173
 libssasmp — Sun Solstice Enterprise SNMP library, 174
 libsys — system library, 175
 libtermcap — screen handling and optimization library, 82
 libtermplib — screen handling and optimization library, 82
 libthread — threads library, 182
 libthread_db — threads debugging library, 186
 libtnfctl — TNF probe control library, 188
 libucb — UCB source compatibility library, 190
 libvolmgt — volume management library, 193
 libvt0 — graphics interface libraries, 144
 libw — wide character library, 194
 libwsreg — product install registry library, 196
 libxfn — X/Open Federated Naming (XFN) library, 198
 libxnet — X/Open Networking library, 203
 liby — yacc library, 206

M

machine-dependent values, — values, 250
 math — math functions and constants, 207
 math functions and constants — math, 207
 memory allocation library — libmalloc, 120
 memory allocator interface library — libbsdmalloc, 49
 menus library — libmenu, 123

multi-threaded memory allocator library — libmtmalloc, 126
 multiple precision library — libmp, 125

N

ndbm — definitions for ndbm database operations, 209
 netdb — definitions for network database operations, 210
 Default, 211
 Standard-conforming, 212
 network services library — libnsl, 127
 nl_types — native language data types, 214

O

obsolete RPC library — librpcoc, 157

P

PAM (Pluggable Authentication Module) library — libpam, 137
 panels library — libpanel, 139
 PICL library — libpicl, 141
 PICL plug-in library — libpicltree, 142
 pool configuration manipulation library — libpool, 146
 POSIX.1b Realtime Extensions library — libposix4, 161
 POSIX.1b Realtime Extensions library — librt, 161
 POSIX threads library — libpthread, 151
 process accounting, — acct, 28
 processes
 base signals — signal, 220
 signal generation information — signinfo, 216
 wait status — wstat, 254
 product install registry library — libwsreg, 196

R

remote asynchronous calls library — librac, 154

remote shared memory interface library —
librsm, 159
resolver library — libresolv, 155
RPC services library — librpcsvc, 158
runtime linker debugging library —
librtld_db, 163

S

screen handling and optimization library
— libcurses, 82
— libtermcap, 82
— libtermlib, 82
security attributes database library —
libsecdb, 165
sendfile library — libsendfile, 167
service location protocol library — libslp, 168
signal — base signals, 220
signal generation information, — siginfo, 216
smartcard library — libsmartcard, 169
socket — Internet Protocol family, 227
sockets library — libsocket, 171
standard symbolic constants and types —
unistd, 243
stat — data returned by stat system call, 232
string pattern-matching library — libgen, 106
Sun Solstice Enterprise Agent Component
library — libdmici, 97
Sun Solstice Enterprise Agent DMI library —
libdmi, 96
Sun Solstice Enterprise Agent library —
libssagent, 173
Sun Solstice Enterprise Agent Management
library — libdmimi, 98
Sun Solstice Enterprise SNMP library —
libssasmp, 174
system calls, — stat, 232
system library — libsys, 175

T

threads debugging library — libthread_db, 186
threads library — libthread, 182
TNF probe control library — libtnfctl, 188
types32 — fixed-width data types, 238

U

the UCB compatibility library — libucb, 190
un — definitions for UNIX-domain sockets, 242
unistd — standard symbolic constants and
types, 243
UNIX System Code, data types — types, 239
user context, — ucontext, 241
user mailbox lockfile management library —
libmail, 119

V

values — machine-dependent values, 250
variable arguments
handle list — stdarg, 234
handle list — vararg, 252
volume management library — libvolmgt, 193

W

wait status, — wstat, 254
wide character library — libw, 194

X

X/Open Federated Naming (XFN) library —
libxfn, 198
X/Open Networking library — libxnet, 203

Y

yacc library — liby, 206